

# ***TMS320C54x DSP***

## **高性能ペリフェラル**

# **リファレンス・セット**

## *Volume.5*

***TMS320C54x DSP***  
高機能ペリフェラル  
**リファレンスセット** Volume. 5

1999年12月



## ご 注 意

日本テキサス・インスツルメンツ株式会社（以下TIといいます）は、通知をすることなくその製品を変更し、もしくは半導体集積回路製品またはサービスの製造または提供を中止することがありますので、お客様は、発注される前に、これから参照しようとする資料が最新のものであることを確実にするため、最新版の資料を取得するようお勧めします。

TIは、その半導体集積回路製品および関連するソフトウェアが、TIの標準保証条件に従い販売の際の現行の仕様書に対応した性能を有していることを保証します。検査およびその他の品質管理技法は、TIが当該保証を支援するのに必要とみなす範囲で行なわれております。各デバイスの全てのパラメーターに関する特定の検査は、政府がそれ等の実行を義務づけている場合を除き、必ずしも行なわれておりません。

半導体集積回路製品を使用する或る種の用途の中には、死亡、傷害、または財産もしくは環境に深刻な被害をもたらす危険の可能性を包含するものがあります。（以下、これらを「重大用途」といいます。）

TIの半導体集積回路製品は、生命維持の用途、装置、システム、その他の重大用途に使用できるように設計も、意図も、承認も、また保証もされていません。

TIの製品を当該重大用途に組込むことは、お客様独自のリスクでなされることと解釈されます。TI製品を当該用途に使用される場合は、事前にTIの役員の書面による承諾を必要とします。危険な可能性を有する用途に関する質問は、TIの営業所を通じて、TI迄お寄せ下さい。

お客様の用途にTI製品を使用することに伴う危険を最小のものとするため、製品固有の危険性を最小にするための、適切な設計上および作動する上での安全対策は、お客様がとらなくてはなりません。

TIは製品の使用用途に関する支援、お客様の製品の設計、ソフトウェアの性能、または特許侵害もしくはサービスに対する責任を負うものではありません。またTIは、その半導体集積回路製品もしくはサービスが使用されうる、もしくは使用されている組み合わせ、機械装置、もしくは方法をカバーしている、またはそれ等に関連している特許権、著作権、回路配置利用権、その他の知的財産権に基づいて何らかのライセンスを許諾するということは明示的にも黙示的にも保証も表示もしていません。

Copyright ©1999 日本テキサス・インスツルメンツ株式会社

### 弊社半導体製品の取り扱い・保管について

IN-9909

半導体製品は、取り扱い、保管・輸送環境、基板実装条件によっては、お客様での実装前後に破壊/劣化、または故障を起こすことがあります。

弊社半導体製品のお取り扱い、ご使用にあたっては下記の点を遵守して下さい。

#### 1. 静電気

- 素手で半導体製品単体を触らないこと。どうしても触る必要がある場合は、リストストラップ等で人体からアースをとり、導電性手袋等をして取り扱うこと。
- 弊社出荷梱包単位（外装から取り出された内装及び個装）又は製品単品で取り扱いを行う場合は、接地された導電性のテーブル上で（導電性マットにアースをとったもの等）、アースをした作業者が行うこと。また、コンテナ等も、導電性のものを使うこと。
- マウンタやはんだ付け設備等、半導体の実装に関わる全ての装置類は、静電気の帯電を防止する措置を施すこと。
- 前記のリストストラップ・導電性手袋・テーブル表面及び実装装置類の接地等の静電気帯電防止措置は、常に管理されその機能が確認されていること。

#### 2. 温・湿度環境

- 温度：0～40℃、相対湿度：40～85%で保管・輸送及び取り扱いを行うこと。（但し、露結しないこと。）

- 直射日光が当たる状態で保管・輸送しないこと。

#### 3. 防湿梱包

- 防湿梱包品は、開封後は個別推奨保管環境及び期間に従い基板実装すること。

#### 4. 機械的衝撃

- 梱包品（外装、内装、個装）及び製品単品を落下させたり、衝撃を与えないこと。

#### 5. 熱衝撃

- はんだ付け時は、最低限260℃以上の高温状態に、10秒以上さらさないこと。（個別推奨条件がある時はそれに従うこと。）

#### 6. 汚染

- はんだ付け性を損なう、又はアルミ配線腐食の原因となるような汚染物質（硫黄、塩素等ハロゲン）のある環境で保管・輸送しないこと。
- はんだ付け後は十分にフラックスの洗浄を行うこと。（不純物含有率が一定以下に保証された無洗浄タイプのフラックスは除く。）

以上

# はじめに

---

---

---

### 本書について

TMS320C54xは、テキサス・インスツルメンツ(TI™)が提供するTMS320ファミリーの固定小数点デジタル・シグナル・プロセッサ(DSP)です。これらのデバイスは、省電力の高機能アーキテクチャ・コアを特徴としています。本書(全5巻マニュアルセットの第5巻)では、これらの一部のデバイスで使用可能な高機能ペリフェラルについて説明します。

多くのデバイス・リファレンスは、英数字で記した通常の接頭文字の代わりにアポストロフィ(')を付けて記載されています(たとえば、TMS320VC5420の代わりに'C5420と記載します)。特に指定がない限り、本書で'C54xと表すリファレンスはすべてTMS320VC54xを指します。

このユーザーズ・ガイドでは、'C5402、'C5410、'C5420のデバイスで使用可能な高機能ペリフェラルと、それらの操作方法について説明します。

本書で説明するおもな項目を次に示します。

- ☐ ホスト・ポート・インターフェイス(HPI)
- ☐ マルチチャネル・バッファド・シリアル・ポート(McBSP)
- ☐ DMAコントローラ(DMA)
- ☐ 拡張外部入出力インターフェイス(EnhXIO)
- ☐ 内部でのプロセッサ間FIFO

'C54xデバイスに関連する、従来から搭載されているその他のペリフェラルの詳細については、「テキサス・インスツルメンツ提供の関連マニュアル」節の『TMS320C54x DSP, CPU and Peripherals, Volume 1』を参照してください。

### 表記上の規約

本書では、次の規約が使われています。

- ❑ TMS320C54x DSPでは、ニーモニックか代数表記アセンブラのいずれかの命令セットを使用できます。本書では、ニーモニックの命令セットが使われています。ニーモニックの命令セットについては、『TMS320C54x DSP Reference Set, Volume 2: Mnemonic Instruction Set』を参照してください。代数表記アセンブラの命令セットについては、『TMS320C54x DSP Reference Set, Volume 3: Algebraic Instruction Set』を参照してください。

- ❑ プログラム・リストとプログラム例は、特殊なフォントで示します。

プログラム・リストの一例を次に示します。

```
STL      A,*AR1+      ;Int_RAM(I)=0
RSBX     INTM          ;Globally enable interrupts
B        MAIN_PG      ;Return to foreground program
```

- ❑ 大カッコ([ ])は、オプションのパラメータを示します。オプションのパラメータを使用する場合はカッコ内の情報のみを指定し、カッコ自体は入力しないでください。

### 注記および警告の説明

本書には次のような警告が表記されています。

これは警告文の見本です。

警告文は、ソフトウェアや機器を破損する危険がある状況について説明するものです。

警告に記載された情報は、お客様を保護するためのものです。各警告を注意深くお読みください。

## テキサス・インスツルメンツ提供の関連マニュアル

次に挙げるのは、TMS320C54xの関連マニュアルです。

これらのマニュアルの多くは、インターネット上(<http://www.tij.co.jp>)でご覧になれます。

### **TMS320C54x DSP Reference Set, Volume 1: CPU and Peripherals** (文献番号 SPRU131)

16ビット固定小数点汎用デジタル・シグナル・プロセッサTMS320C54xについての説明書です。ここでは、プロセッサのアーキテクチャ、内部レジスタ構造、データおよびプログラムのアドレッシング、命令のパイプライン、オンチップ・ペリフェラルについて解説しています。また、開発サポート情報、部品リスト、XDS510エミュレータを使用する際の設計に関する考慮事項も記載されています。

### **TMS320C54x CPU及びペリフェラル リファレンス・セット Volume.1**(文献番号 SCJ2287)

TMS320C54xデジタル・シグナル・プロセッサのCPUコアとペリフェラルのユーザーズ・ガイドの日本語版です。上記の英語版(文献番号SPRU131)を翻訳したものです。

### **TMS320C54x DSP Reference Set, Volume 2: Mnemonic Instruction Set** (文献番号SPRU172)

TMS320C54xデジタル・シグナル・プロセッサで使われる各ニーモニック命令についての説明書です。命令セットのクラスとサイクル数についても解説されています。

### **TMS320C54x DSP Reference Set, Volume 3: Algebraic Instruction Set** (文献番号SPRU179)

TMS320C54xデジタル・シグナル・プロセッサで使われる各代数表記命令についての説明書です。命令セットのクラスとサイクル数についても解説されています。

### **TMS320C54x DSP Reference Set, Volume 4: Applications Guide** (文献番号SPRU173)

TMS320C54xデジタル・シグナル・プロセッサ用のソフトウェアおよびハードウェア・アプリケーションについての説明書です。ブートローダ、開発サポート情報、部品リスト、XDS510エミュレータを使用する際の設計に関する考慮事項も記載されています。

### **TMS320C54x DSP Reference Set, Volume 5: Enhanced Peripherals** (文献番号SPRU302)

TMS320C54xデジタル・シグナル・プロセッサで使用可能な高機能ペリフェラルについての説明書です。マルチチャネル・バッファド・シリアル・ポート(McBSP)、ダイレクト・メモリ・アクセス(DMA)コントローラ、HPI-8およびHPI-16ホスト・ポート・インターフェイス、内部でのプロセッサ間通信についても解説しています。

**TMS320C54x 高性能ペリフェラル リファレンス・セット Volume.5** (文献番号SPRU362)  
TMS320C54xデジタル・シグナル・プロセッサの高性能ペリフェラルのユーザーズ・ガイドの日本語版です。上記の英語版(文献番号SPRU302)を翻訳したものです。

**TMS320C54x, TMS320LC54x, TMS320VC54x Fixed-Point Digital Signal Processors** (文献番号SPRS039)  
デバイスの電氣的仕様とタイミングに関する仕様、並びにそのパッケージで使われている信号の説明とピン配置が記載されたデータ・シートです。

**TMS320C54x DSKplus User's Guide** (文献番号SPRU191)  
TMS320C54xデジタル・シグナル・プロセッサのスタータ・キット(DSK)についての説明書です。DSKを使用すると、'C54xコードをリアルタイムで実行し、これを1行ずつデバッグすることができます。ここでは、インストール手順、デバッグとアセンブラ、アプリケーションのカスタマイズ、初期化ルーチンについて解説されています。

**TMS320C54x DSKplus ユーザーズ・ガイド** (文献番号SCJ2288)  
TMS320C54xデジタル・シグナル・プロセッサのスタータ・キット(DSK)のユーザーズ・ガイドの日本語版。上記の英語版(文献番号SPRU191)を翻訳したものです。

**TMS320C54x Assembly Language Tools User's Guide** (文献番号SPRU102)  
このマニュアルでは、'C54xで使われるアセンブリ言語ツール(アセンブラやリンカなどのアセンブリ言語コード開発ツール)、アセンブラの疑似命令、マクロ、COFF疑似命令、シンボル形式のデバッグ疑似命令について説明しています。

**TMS320C5xx C Source Debugger User's Guide** (文献番号SPRU099)  
'C54xのエミュレータ、評価モジュール、シミュレータについての説明書です。ここでは、ウィンドウ管理、コマンド入力、コード実行、データ管理、ブレークポイントなど、デバッグ・インターフェイスのさまざまな項目について解説しています。また、デバッグの基本機能を紹介するチュートリアルも含まれています。

**TMS320C54x Code Generation Tools Getting Started Guide** (文献番号SPRU147)  
TMS320C54xアセンブリ言語ツールと'C54xデバイス用Cコンパイラのインストール方法についての説明書です。

**TMS320C54x Evaluation Module Technical Reference** (文献番号SPRU135)  
'C54x評価モジュール、モジュールの各種機能、設計の詳細、および外部インターフェイスについての説明書です。

**TMS320C54x Optimizing C Compiler User's Guide** (文献番号SPRU103)  
'C54xのCコンパイラについての説明書です。このCコンパイラは、ANSI規格のCソースコードを受け入れ、'C54x世代のデバイスに使われるTMS320アセンブリ言語ソースコードを生成します。

**TMS320C54x Simulator Getting Started** (文献番号SPRU137)  
TMS320C54xシミュレータと'C54x用Cソース・デバッグのインストール方法についての説明書です。

**TMS320 Third-Party Support Reference Guide** (文献番号SPRU052)

TMS320デジタル・シグナル・プロセッサ・ファミリー対応の各種製品を提供している100社以上のサードパーティーを、アルファベット順に一覧したものです。ソフトウェアおよびハードウェア開発ツール、音声認識、画像処理、ノイズ・キャンセレーション、モデムなど、極めて多数の製品とアプリケーションが紹介されています。

**TMS320C548/C549 Bootloader Technical Reference** (文献番号SPRU288)

このマニュアルでは、C548及びC549の電源投入時にブートローダが外部ソースからプログラム・メモリーにユーザー・コードを転送する際の処理について解説しています(現在のところインターネット上でしかご覧になれません)。

## 技術資料

デジタル信号処理については、幅広い関連文書が出されています。ここで紹介する参考文献は、次の応用分野のいずれかに該当しています。

- ☐ 汎用DSP
- ☐ 画像処理
- ☐ スピーチ/音声
- ☐ 制御
- ☐ マルチメディア
- ☐ 軍需
- ☐ 情報通信
- ☐ 自動車
- ☐ 民生機器
- ☐ 医療
- ☐ 開発サポート

次のリストでは、参考文献を著者名のアルファベット順に示します。これらの文献には、信号処理システムの設計、動作、および応用法に関する有用情報が記載されています。これらの文献はすべて補足的な参考として役立ちます。テキサス・インスツルメンツでは、これらの出版物を参照することを強くお勧めします。

### 汎用DSP

- 1) Chassaing, R./Horning, D.W.編『*Digital Signal Processing with Fixed and Floating-Point Processors*』米国、第1巻、第1号、1~4ページ、1991年3月
- 2) Defatta, David J./Joseph G. Lucas/William S. Hodgkiss共著『*Digital Signal Processing: A System Design Approach*』New York: John Wiley刊、1988年



- 3) Erskine, C./S. Magar共著「Architecture and Applications of a Second-Generation Digital Signal Processor」『*Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*』米国、1985年
- 4) Essig, D./C. Erskine/E. Caudel/S. Magar共著「A Second-Generation Digital Signal Processor」『*IEEE Journal of Solid-State Circuits*』米国、第SC-21巻、第1号、86～91ページ、1986年2月
- 5) Frantz, G./K. Lin/J. Reimer/J. Bradley共著「The Texas Instruments TMS320C25 Digital Signal Microcomputer」『*IEEE Microelectronics*』米国、第6巻、第6号、10～28ページ、1986年12月
- 6) Gass, W./R. Tarrant/T. Richard/B. Pawate/M. Gammel/P. Rajasekaran/R. Wiggins/C. Covington共著「Multiple Digital Signal Processor Environment for Intelligent Signal Processing」『*Proceedings of the IEEE*』米国、第75巻、第9号、1246～1259ページ、1987年9月
- 7) Jackson, Leland B.著『*Digital Filters and Signal Processing*』Hingham, MA: Kluwer Academic Publishers刊、1986年
- 8) Jones, D.L./T.W. Parks共著『*A Digital Signal Processing Laboratory Using the TMS32010*』Englewood Cliffs, NJ: Prentice-Hall, Inc.刊、1987年
- 9) Lim, Jae/Alan V. Oppenheim共著『*Advanced Topics in Signal Processing*』Englewood Cliffs, NJ: Prentice-Hall, Inc.刊、1988年
- 10) Lin, K./G. Frantz/R. Simar, Jr.共著「The TMS320 Family of Digital Signal Processors」『*Proceedings of the IEEE*』米国、第75巻、第9号、1143～1159ページ、1987年9月
- 11) Lovrich, A./Reimer, J.共著「An Advanced Audio Signal Processor」『*Digest of Technical Papers for 1991 International Conference on Consumer Electronics*』1991年6月
- 12) Magar, S./D. Essig/E. Caudel/S. Marshall/R. Peters共著「An NMOS Digital Signal Processor with Multiprocessing Capability」『*Digest of IEEE International Solid-State Circuits Conference*』米国、1985年2月
- 13) Oppenheim, Alan V./R.W. Schaffer共著『*Digital Signal Processing*』Englewood Cliffs, NJ: Prentice-Hall, Inc.刊、1975/1988年
- 14) Papamichalis, P.E./C.S. Burrus共著「Conversion of Digit-Reversed to Bit-Reversed Order in FFT Algorithms」『*Proceedings of ICASSP89*』米国、984～987ページ、1989年5月
- 15) Papamichalis, P./R. Simar, Jr.共著「The TMS320C30 Floating-Point Digital Signal Processor」『*IEEE Micro Magazine*』米国、13～29ページ、1988年12月

- 16) Papamichalis, P.E.著「FFT Implementation on the TMS320C30」『*Proceedings of ICASSP 88*』米国、第D巻、1399ページ、1988年4月
- 17) Parks, T.W./C.S. Burrus共著『*Digital Filter Design*』New York, NY: John Wiley and Sons, Inc.刊、1987年
- 18) Peterson, C./Zervakis, M./Shehadeh, N.共著「Adaptive Filter Design and Implementation Using the TMS320C25 Microprocessor」『*Computers in Education Journal*』米国、第3巻、第3号、12～16ページ、1993年6～9月
- 19) Prado, J./R. Alcantara共著「A Fast Square-Rooting Algorithm Using a Digital Signal Processor」『*Proceedings of IEEE*』米国、第75巻、第2号、262～264ページ、1987年2月
- 20) Rabiner, L.R/B. Gold共著『*Theory and Applications of Digital Signal Processing*』Englewood Cliffs, NJ: Prentice-Hall, Inc.刊、1975年
- 21) Simar, Jr., R./A. Davis共著「The Application of High-Level Languages to Single-Chip Digital Signal Processors」『*Proceedings of ICASSP 88*』米国、第D巻、1678ページ、1988年4月
- 22) Simar, Jr., R./T. Leigh/P. Koeppen/J. Leach/J. Potts/D. Blalock共著「A 40MFLOPS Digital Signal Processor: the First Supercomputer on a Chip」『*Proceedings of ICASSP 87*』米国、カタログ番号87CH2396-0、第1巻、535～538ページ、1987年4月
- 23) Simar, Jr., R./J. Reimer共著「The TMS320C25: a 100 ns CMOS VLSI Digital Signal Processor」『*1986 Workshop on Applications of Signal Processing to Audio and Acoustics*』1986年9月
- 24) Texas Instruments編『*Digital Signal Processing Applications with the TMS320 Family, 1986*』Englewood Cliffs, NJ: Prentice-Hall, Inc.刊、1987年
- 25) Treichler, J.R./C.R. Johnson, Jr./M.G. Larimore共著『*A Practical Guide to Adaptive Filter Design*』New York, NY: John Wiley and Sons, Inc.刊、1987年

## 画像処理

- 1) Reimer, J./A. Lovrich共著「Graphics with the TMS32020」『*WESCON/85 Conference Record*』米国、1985年

## スピーチ/音声

- 1) DellaMorte, J./P. Papamichalis共著「Full-Duplex Real-Time Implementation of the FED-STD-1015 LPC-10e Standard V.52 on the TMS320C25」『*Proceedings of SPEECH TECH 89*』218～221ページ、1989年5月
- 2) Gray, A.H./J.D. Markel共著『*Linear Prediction of Speech*』New York, NY: Springer-Verlag刊、1976年
- 3) Frantz, G.A./K.S. Lin共著「A Low-Cost Speech System Using the TMS320C17」『*Proceedings of SPEECH TECH '87*』25～29ページ、1987年4月
- 4) Papamichalis, P./D. Lively共著「Implementation of the DOD Standard LPC-10/52E on the TMS320C25」『*Proceedings of SPEECH TECH '87*』201～204ページ、1987年4月
- 5) Papamichalis, Panos著『*Practical Approaches to Speech Coding*』Englewood Cliffs, NJ: Prentice-Hall, Inc.刊、1987年
- 6) Pawate, B.I./G.R. Doddington共著「Implementation of a Hidden Markov Model-Based Layered Grammar Recognizer」『*Proceedings of ICASSP 89*』米国、801～804ページ、1989年5月
- 7) Rabiner, L.R./R.W. Schafer共著『*Digital Processing of Speech Signals*』Englewood Cliffs, NJ: Prentice-Hall, Inc.刊、1978年
- 8) Reimer, J.B./K.S. Lin,共著「TMS320 Digital Signal Processors in Speech Applications」『*Proceedings of SPEECH TECH '88*』1988年4月
- 9) Reimer, J.B./M.L. McMahan/W.W. Anderson共著「Speech Recognition for a Low-Cost System Using a DSP」『*Digest of Technical Papers for 1987 International Conference on Consumer Electronics*』1987年6月

## 制御

- 1) Ahmed, I.著「16-Bit DSP Microcontroller Fits Motion Control System Application」『*PCIM*』、1988年6月
- 2) Ahmed, I.著「Implementation of Self Tuning Regulators with TMS320 Family of Digital Signal Processors」『*MOTORCON '88*』248～262ページ、1988年9月
- 3) Ahmed, I./S. Lindquist共著「Digital Signal Processors: Simplifying High-Performance Control」『*Machine Design*』1987年9月
- 4) Ahmed, I./S. Meshkat共著「Using DSPs in Control」『*Control Engineering*』1988年2月

- 5) Allen, C/P Pillay共著「TMS320 Design for Vector and Current Control of AC Motor Drives」『Electronics Letters』英国、第28巻、第23号、2188～2190ページ、1992年11月
- 6) Bose, B.K./P.M. Szczesny共著「A Microcomputer-Based Control and Simulation of an Advanced IPM Synchronous Machine Drive System for Electric Vehicle Propulsion」『Proceedings of IECON '87』第1巻、454～463ページ、1987年11月
- 7) Hanselman, H.著「LQG-Control of a Highly Resonant Disc Drive Head Positioning Actuator」『IEEE TransactionsonIndustrialElectronics』米国、第35巻、第1号、100～104ページ、1988年2月
- 8) Lovrich, A./G. Troullinos/R. Chirayil共著「An All-Digital Automatic Gain Control」『Proceedings of ICASSP 88』米国、第D巻、1734ページ、1988年4月
- 9) Matsui, N./M. Shigyo共著「Brushless DC Motor Control Without Position and Speed Sensors」『IEEE Transactions on Industry Applications』米国、第28巻、第1号、第1部、120～127ページ、1992年1～2月
- 10) Meshkat, S./I. Ahmed共著「Using DSPs in AC Induction Motor Drives」『Control Engineering』1988年2月
- 11) Panahi, I./R. Restle共著「DSPs Redefine Motion Control」『Motion Control Magazine』1993年12月

#### マルチメディア

- 1) Reimer, J.著「DSP-Based Multimedia Solutions Lead Way Enhancing Audio Compression Performance」『Dr. Dobbs Journal』1993年12月
- 2) Reimer, J./G. Benbassat/W. Bonneau Jr.共著「Application Processors: Making PC Multimedia Happen」『Silicon Valley PC Design Conference』1991年7月

#### 軍備

- 1) Papamichalis, P./J. Reimer共著「Implementation of the Data Encryption Standard Using the TMS32010」『Digital Signal Processing Applications』1986年

## 情報通信

- 1) Ahmed, I./A. Lovrich共著「Adaptive Line Enhancer Using the TMS320C25」『*Conference Records of Northcon/86*』米国、14/3/1～10、1986年9～10月
- 2) Casale, S./R. Russo/G. Bellina共著「Optimal Architectural Solution Using DSP Processors for the Implementation of an ADPCM Transcoder」『*Proceedings of GLOBECOM '89*』1267～1273ページ、1989年11月
- 3) Cole, C./A. Haoui/P. Winship共著「A High-Performance Digital Voice Echo Canceller on a SINGLE TMS32020」『*Proceedings of ICASSP 86*』米国、カタログ番号 86CH2243-4、第1巻、429～432ページ、1986年4月
- 4) Cole, C./A. Haoui/P. Winship共著「A High-Performance Digital Voice Echo Canceller on a Single TMS32020」『*Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*』米国、1986年
- 5) Lovrich, A./J. Reimer共著「A Multi-Rate Transcoder」『*Transactions on Consumer Electronics*』米国、1989年11月
- 6) Lovrich, A./J. Reimer共著「A Multi-Rate Transcoder」『*Digest of Technical Papers for 1989 International Conference on Consumer Electronics*』1989年6月7～9日
- 7) Lu, H./D. Hedberg/B. Fraenkel共著「Implementation of High-Speed Voiceband Data Modems Using the TMS320C25」『*Proceedings of ICASSP 87*』米国、カタログ番号 87CH2396～0、第4巻、1915～1918ページ、1987年4月
- 8) Mock, P.著「Add DTMF Generation and Decoding to DSP- $\mu$ P Designs」『*Electronic Design*』米国、第30巻、第6号、205～213ページ、1985年3月
- 9) Reimer, J./M. McMahan/M. Arjmand共著「ADPCM on a TMS320 DSP Chip」『*Proceedings of SPEECH TECH 85*』246～249ページ、1985年4月
- 10) Troullos, G./J. Bradley共著「Split-Band Modem Implementation Using the TMS32010 Digital Signal Processor」『*Conference Records of Electro/86 and Mini/Micro Northeast*』米国、14/1/1～21、1986年5月

## 自動車

- 1) Lin, K.著「Trends of Digital Signal Processing in Automotive」『*International Congress on Transportation Electronic (CONVERGENCE '88)*』1988年10月

## 民生機器

- 1) Frantz, G.A./J.B. Reimer/R.A. Wotiz共著「Julie, The Application of DSP to a Product」  
『*Speech Tech Magazine*』 米国、1988年9月
- 2) Reimer, J.B./G.A. Frantz共著「Customization of a DSP Integrated Circuit for a Customer Product」  
『*Transactions on Consumer Electronics*』 米国、1988年8月
- 3) Reimer, J.B./P.E. Nixon/E.B. Boles/G.A. Frantz共著「Audio Customization of a DSP IC」  
『*Digest of Technical Papers for 1988 International Conference on Consumer Electronics*』 1988年6月8～10日

## 医療

- 1) Knapp/Townshend編「A Real-Time Digital Signal Processing System for an Auditory Prosthesis」  
『*Proceedings of ICASSP 88*』 米国、第A巻、2493ページ、1988年4月
- 2) Morris, L.R./P.B. Barszczewski共著「Design and Evolution of a Pocket-Sized DSP Speech Processing System for a Cochlear Implant and Other Hearing Prosthesis Applications」  
『*Proceedings of ICASSP 88*』 米国、第A巻、2516ページ、1988年4月

## 開発サポート

- 1) Mersereau, R./R. Schafer/T. Barnwell/D. Smith共著「A Digital Filter Design Package for PCs and TMS320」  
『*MIDCON/84 Electronic Show and Convention*』 米国、1984年
- 2) Simar, Jr., R./A. Davis共著「The Application of High-Level Languages to Single-Chip Digital Signal Processors」  
『*Proceedings of ICASSP 88*』 米国、第3巻、1678～1681ページ、1988年4月

## 商標

---

### 商標

MS-DOSはMicrosoft Corporationの登録商標です。

OS/2およびPC-DOSはInternational Business Machines Corporationの商標です。

PAL<sup>®</sup>はAdvanced Micro Devices, Inc.の登録商標です。

WindowsはMicrosoft Corporationの登録商標です。

TI、XDS510、XDS510WSはTexas Instruments Incorporatedの商標です。

Micro StarはTexas Instruments Incorporatedの商標です。

SPIはMotorola Corporationの商標です。

# 目次

|       |                                                    |      |
|-------|----------------------------------------------------|------|
| 1     | はじめに .....                                         | 1-1  |
|       | 'C5402, 'C5410, 'C5420で使用可能な高機能ペリフェラルの概要を紹介しています。  |      |
| 1.1   | 'C54x高機能ペリフェラルの概要 .....                            | 1-2  |
| 1.1.1 | マルチチャネル・バッファド・シリアル・ポート(McBSP) .....                | 1-2  |
| 1.1.2 | ダイレクト・メモリー・アクセス(DMA:Direct Memory Access)コントローラ .. | 1-2  |
| 1.1.3 | ホスト・ポート・インターフェイス(HPI-8およびHPI-16) .....             | 1-2  |
| 2     | マルチチャネル・バッファド・シリアル・ポート(McBSP) .....                | 2-1  |
|       | マルチチャネル・バッファド・シリアル・ポートの機能と動作を説明しています。              |      |
| 2.1   | McBSPの特長 .....                                     | 2-2  |
| 2.2   | McBSPの概要 .....                                     | 2-3  |
| 2.2.1 | シリアル・ポートの設定 .....                                  | 2-6  |
| 2.2.2 | 受信制御レジスタおよび送信制御レジスタ: RCR[1,2]およびXCR[1,2] .....     | 2-15 |
| 2.3   | データ送受信フロー .....                                    | 2-22 |
| 2.3.1 | シリアル・ポートのリセット: (R/X)RST、およびRESET .....             | 2-22 |
| 2.3.2 | レディー状態の決定 .....                                    | 2-26 |
| 2.3.3 | CPU割り込み: (R/X)INT .....                            | 2-27 |
| 2.3.4 | フレームおよびクロックの設定 .....                               | 2-27 |
| 2.3.5 | McBSPの基本的な動作 .....                                 | 2-38 |
| 2.3.6 | フレーム同期の無視 .....                                    | 2-40 |
| 2.3.7 | シリアル・ポートの例外条件 .....                                | 2-43 |
| 2.3.8 | 受信データの位置調整と符号拡張: RJUST .....                       | 2-52 |
| 2.4   | μ 則/A則圧伸ハードウェアの動作: (R/X)COMPAND .....              | 2-53 |
| 2.4.1 | 内部データの圧伸 .....                                     | 2-55 |
| 2.5   | プログラマブル・クロックおよびフレーム生成 .....                        | 2-57 |
| 2.5.1 | サンプル・レート生成器のクロックとフレームの生成 .....                     | 2-58 |
| 2.5.2 | データ・クロックの生成 .....                                  | 2-62 |
| 2.5.3 | フレーム同期信号の生成 .....                                  | 2-66 |
| 2.5.4 | クロック・タイミングの例 .....                                 | 2-69 |
| 2.6   | マルチチャネル選択動作 .....                                  | 2-72 |
| 2.6.1 | マルチチャネル動作の制御レジスタ .....                             | 2-73 |
| 2.6.2 | マルチチャネル選択のイネーブル .....                              | 2-76 |
| 2.6.3 | チャネルのマスクとイネーブル .....                               | 2-76 |
| 2.6.4 | A-bisインターフェイス機能('C5410のみで使用可能) .....               | 2-83 |



## 目次

|       |                                                  |      |
|-------|--------------------------------------------------|------|
| 2.7   | SPIプロトコル: McBSPクロック・ストップ・モード                     | 2-86 |
| 2.7.1 | クロック・ストップ・モードの設定と信号の説明                           | 2-88 |
| 2.7.2 | SPIマスタとしてのMcBSP動作                                | 2-90 |
| 2.7.3 | SPIスレーブとしてのMcBSP動作                               | 2-92 |
| 2.7.4 | SPIモードでのMcBSP初期化                                 | 2-93 |
| 2.8   | エミュレーションFREEビットおよびSOFTビット                        | 2-95 |
| 2.9   | McBSPの汎用入出力ポート                                   | 2-96 |
| 2.10  | パワーダウン・モードにおけるMcBSPの動作                           | 2-98 |
| 2.11  | McBSPプログラミングのサンプル・プログラム                          | 2-99 |
| 3     | ダイレクト・メモリー・アクセス(DMA)コントローラ                       | 3-1  |
|       | ダイレクト・メモリー・アクセス(DMA)コントローラの機能と動作について説明しています。     |      |
| 3.1   | DMAの概要                                           | 3-2  |
| 3.2   | DMAの動作および設定方法                                    | 3-4  |
| 3.2.1 | レジスタのサブアドレッシング                                   | 3-4  |
| 3.2.2 | DMAチャネル優先順位およびイネーブル制御レジスタ                        | 3-7  |
| 3.2.3 | チャネル・コンテキスト・レジスタ                                 | 3-11 |
| 3.3   | 拡張アドレッシング                                        | 3-33 |
| 3.4   | DMAメモリー・マップ                                      | 3-34 |
| 3.4.1 | 'C5402のDMAメモリー・マップ                               | 3-34 |
| 3.4.2 | 'C5410のDMAメモリー・マップ                               | 3-35 |
| 3.4.3 | 'C5420のDMAメモリー・マップ                               | 3-36 |
| 3.5   | DMA転送のレイテンシー                                     | 3-39 |
| 3.6   | DMAコントローラを介した拡張ホスト・ポート・インターフェイスのアクセス             | 3-42 |
| 3.7   | 'C5420の内部でのプロセッサ間FIFOによる通信                       | 3-43 |
| 3.8   | パワーダウン・モードでのDMA動作                                | 3-43 |
| 3.9   | プログラム例                                           | 3-44 |
| 4     | 拡張8ビット・ホスト・ポート・インターフェイス(HPI-8)                   | 4-1  |
|       | 拡張8ビット・ホスト・ポート・インターフェイス(HPI-8)の機能と動作について説明しています。 |      |
| 4.1   | 拡張8ビット・ホスト・ポート・インターフェイス(HPI-8)の概要                | 4-2  |
| 4.2   | HPI-8の基本機能について                                   | 4-4  |
| 4.3   | HPI-8の動作について                                     | 4-6  |
| 4.3.1 | HPI-8のアドレス・レジスタとメモリー・マップ                         | 4-9  |
| 4.3.2 | HPI-8の拡張アドレッシング                                  | 4-10 |
| 4.3.3 | アドレスの自動インクリメント                                   | 4-11 |
| 4.3.4 | HPI-8の制御レジスタ・ビットとその機能                            | 4-12 |
| 4.4   | ホストからHPI-8へのリード/ライト・アクセス                         | 4-14 |
| 4.4.1 | HPI-8アクセスのレイテンシー                                 | 4-16 |
| 4.4.2 | アクセス・シーケンスの例                                     | 4-19 |
| 4.5   | DSPINTとHINTの動作                                   | 4-23 |
| 4.5.1 | DSPINTを使用したホスト・デバイスによる'C54xへの割り込み                | 4-23 |

|          |                                                                                     |            |
|----------|-------------------------------------------------------------------------------------|------------|
| 4.5.2    | HINTを使用した'C54xによるホスト・デバイスへの割り込み                                                     | 4-23       |
| 4.6      | クロック・モード変更時のHPI-8転送に関する考慮事項                                                         | 4-24       |
| 4.7      | IDLE使用時の考慮事項                                                                        | 4-26       |
| 4.7.1    | IDLE1およびIDLE2モードでのHPI-8アクセス                                                         | 4-26       |
| 4.7.2    | IDLE3モードでのHPI-8アクセス                                                                 | 4-26       |
| 4.8      | リセット時におけるHPI-8の動作                                                                   | 4-28       |
| 4.8.1    | リセット後のHPI-8へのアクセス                                                                   | 4-28       |
| 4.8.2    | リセット時のHPI-8へのアクセス('C5410のみ)                                                         | 4-28       |
| 4.9      | HPI-8データ・ピンの汎用I/Oピンとしての使用('C5410では使用不可)                                             | 4-30       |
| 4.9.1    | GPIO機能の使用法                                                                          | 4-36       |
| <b>5</b> | <b>拡張16ビット・ホスト・ポート・インターフェイス(HPI-16)</b>                                             | <b>5-1</b> |
|          | 拡張16ビット・ホスト・ポート・インターフェイス(HPI-16)の機能と動作について説明しています。                                  |            |
| 5.1      | HPI-16の動作の概要                                                                        | 5-2        |
| 5.2      | 共用モード                                                                               | 5-7        |
| 5.2.1    | HASを使用するホスト・アクセス                                                                    | 5-10       |
| 5.2.2    | HASを使用しないホスト・アクセス                                                                   | 5-11       |
| 5.2.3    | 自動インクリメント動作                                                                         | 5-12       |
| 5.3      | 分離モード                                                                               | 5-15       |
| 5.4      | HPI-16メモリー・マップ                                                                      | 5-18       |
| 5.5      | HPI-16とDMAの相互作用                                                                     | 5-19       |
| 5.6      | リセット時のHPI-16の動作('VC5420のみ)                                                          | 5-21       |
| 5.7      | IDLEn時のHPI-16の動作                                                                    | 5-21       |
| 5.8      | DSPクロック・モードの変更によるHPI-16への影響                                                         | 5-22       |
| <b>6</b> | <b>内部でのプロセッサ間通信</b>                                                                 | <b>6-1</b> |
|          | コア間FIFO通信や、EMIF-HPI通信(外部メモリー・インターフェイスとホスト・ポート・インターフェイスとの内部通信)などのマルチコア通信について説明しています。 |            |
| 6.1      | マルチコアDSP内での通信                                                                       | 6-2        |
| 6.2      | 双方向FIFO                                                                             | 6-3        |
| 6.3      | 外部メモリー空間からHPI-16へのアクセス                                                              | 6-7        |
| 6.4      | McBSPを使用するサブシステム通信                                                                  | 6-10       |
| 6.5      | 内部プロセッサ間割り込み                                                                        | 6-11       |
| <b>7</b> | <b>拡張外部バス・インターフェイス(EnhXIO)</b>                                                      | <b>7-1</b> |
|          | 拡張外部パラレル・インターフェイスの強化された機能について説明しています。                                               |            |
| 7.1      | 拡張外部バス・インターフェイス(XIO2)の概要                                                            | 7-2        |
| 7.1.1    | 追加機能                                                                                | 7-2        |
| 7.2      | バス・タイミング                                                                            | 7-3        |
| <b>A</b> | <b>用語集</b>                                                                          | <b>A-1</b> |



|       |                                          |      |
|-------|------------------------------------------|------|
| 2-1.  | McBSPの内部ブロック図 .....                      | 2-3  |
| 2-2.  | シリアル・ポート制御レジスタ(SPCR1) .....              | 2-7  |
| 2-3.  | シリアル・ポート制御レジスタ2(SPCR2) .....             | 2-10 |
| 2-4.  | ピン制御レジスタ(PCR) .....                      | 2-12 |
| 2-5.  | 受信制御レジスタ(RCR1) .....                     | 2-16 |
| 2-6.  | 受信制御レジスタ2(RCR2) .....                    | 2-17 |
| 2-7.  | 送信制御レジスタ1(XCR1) .....                    | 2-19 |
| 2-8.  | 送信制御レジスタ2(XCR2) .....                    | 2-20 |
| 2-9.  | フレームおよびクロックの動作 .....                     | 2-28 |
| 2-10. | 受信データとクロック .....                         | 2-30 |
| 2-11. | デュアル・フェーズ・フレームの例 .....                   | 2-30 |
| 2-12. | 4つの8ビット・ワードで構成されるシングル・フェーズ・フレーム .....    | 2-33 |
| 2-13. | 1つの32ビット・ワードで構成されるシングル・フェーズ・フレーム .....   | 2-34 |
| 2-14. | データ遅延 .....                              | 2-34 |
| 2-15. | フレーミング・ビット破棄に使用される2ビットのデータ遅延 .....       | 2-35 |
| 2-16. | AC97のデュアル・フェーズ・フレーム・フォーマット .....         | 2-36 |
| 2-17. | フレーム同期付近のAC97ビット・タイミングの例 .....           | 2-36 |
| 2-18. | ノーマル・モードにおけるDXイネーブラ .....                | 2-37 |
| 2-19. | A-bisモードでのDXイネーブラ .....                  | 2-37 |
| 2-20. | McBSPの基本的な動作 .....                       | 2-38 |
| 2-21. | 受信動作 .....                               | 2-39 |
| 2-22. | 送信動作 .....                               | 2-39 |
| 2-23. | 最大フレーム周波数での受信/送信((R/X)DATDLY = 1) .....  | 2-40 |
| 2-24. | 8ビット・データでの最大パケット周波数動作 .....              | 2-41 |
| 2-25. | (R/X)FIG=1の場合の最大パケット周波数での効率的なデータ転送 ..... | 2-42 |
| 2-26. | (R/X)FIG = 0での予定外のフレーム同期 .....           | 2-43 |
| 2-27. | (R/X)FIG = 1での予定外のフレーム同期 .....           | 2-43 |
| 2-28. | シリアル・ポート受信オーバーラン .....                   | 2-45 |
| 2-29. | シリアル・ポート受信オーバーランの回避 .....                | 2-46 |
| 2-30. | 受信フレーム同期パルスへの応答 .....                    | 2-47 |
| 2-31. | 予定外の受信同期パルス .....                        | 2-48 |
| 2-32. | データ上書きをともなう送信 .....                      | 2-48 |
| 2-33. | 送信エンプティ .....                            | 2-49 |
| 2-34. | 送信エンプティの回避 .....                         | 2-50 |
| 2-35. | 送信フレーム同期への応答 .....                       | 2-51 |
| 2-36. | 予定外の送信フレーム同期パルス .....                    | 2-52 |

|       |                                         |      |
|-------|-----------------------------------------|------|
| 2-37. | 圧伸フロー                                   | 2-54 |
| 2-38. | μ則送信データ圧伸フォーマット                         | 2-54 |
| 2-39. | A則送信データ圧伸フォーマット                         | 2-54 |
| 2-40. | 内部データの圧伸                                | 2-55 |
| 2-41. | クロックおよびフレーム生成                           | 2-57 |
| 2-42. | サンプル・レート生成器                             | 2-58 |
| 2-43. | サンプル・レート生成器レジスタ1(SRGR1)                 | 2-59 |
| 2-44. | サンプル・レート生成器レジスタ2(SRGR2)                 | 2-60 |
| 2-45. | GSYNC = 1かつCLKGDV = 1の場合のCLKG同期およびFSG生成 | 2-63 |
| 2-46. | GSYNC = 1かつCLKGDV = 3の場合のCLKG同期およびFSG生成 | 2-64 |
| 2-47. | プログラム可能なフレーム周期とフレーム幅                    | 2-67 |
| 2-48. | STバスとMVIPの例                             | 2-69 |
| 2-49. | シングルレート・クロックの例                          | 2-70 |
| 2-50. | ダブルレート・クロックの例                           | 2-71 |
| 2-51. | マルチチャネル制御レジスタ1(MCR1)                    | 2-73 |
| 2-52. | マルチチャネル制御レジスタ2(MCR2)                    | 2-75 |
| 2-53. | パーティションAおよびBにおけるブロックによるチャネルのイネーブル       | 2-77 |
| 2-54. | XMCMの動作                                 | 2-79 |
| 2-55. | 受信チャネル・イネーブル・レジスタ・パーティションA(RCERA)       | 2-80 |
| 2-56. | 受信チャネル・イネーブル・レジスタ・パーティションB(RCERB)       | 2-81 |
| 2-57. | 送信チャネル・イネーブル・レジスタ・パーティションA(XCERA)       | 2-81 |
| 2-58. | 送信チャネル・イネーブル・レジスタ・パーティションB(XCERB)       | 2-82 |
| 2-59. | A-bisモードの受信動作                           | 2-84 |
| 2-60. | A-bisモードの送信動作(zはハイ・インピーダンスを示す)          | 2-84 |
| 2-61. | 一般的なSPIインターフェイス                         | 2-86 |
| 2-62. | SPI構成: マスタとしてのMcBSP                     | 2-87 |
| 2-63. | SPI構成: スレーブとしてのMcBSP                    | 2-87 |
| 2-64. | CLKSTP = 10bおよびCLKXP = 0でのSPI転送         | 2-88 |
| 2-65. | CLKSTP = 11b、CLKXP = 0でのSPI転送           | 2-89 |
| 2-66. | CLKSTP = 10b、CLKXP = 1でのSPI転送           | 2-89 |
| 2-67. | CLKSTP = 11b、CLKXP = 1でのSPI転送           | 2-89 |
| 3-1.  | レジスタのサブアドレッシング                          | 3-4  |
| 3-2.  | DMAチャネル優先順位およびイネーブル制御(DMPREC)レジスタ       | 3-7  |
| 3-3.  | DMA同期イベントおよびフレーム・カウント(DMSFCn)レジスタ       | 3-13 |
| 3-4.  | DMA転送モード制御(DMMCRn)レジスタ                  | 3-17 |
| 3-5.  | データのソート例                                | 3-23 |
| 3-6.  | DMA転送元プログラム・ページ・アドレス・レジスタ(DMSRCP)       | 3-33 |
| 3-7.  | DMA転送先プログラム・ページ・アドレス・レジスタ(DMDSTP)       | 3-33 |
| 4-1.  | ホスト・ポート・インターフェイスのブロック図                  | 4-3  |
| 4-2.  | 一般的なシステム・ブロック図                          | 4-4  |
| 4-3.  | HPIストローブと選択ロジック                         | 4-8  |
| 4-4.  | HPI-8メモリー・マップ                           | 4-9  |

|       |                                            |      |
|-------|--------------------------------------------|------|
| 4-5.  | HPICレジスタ – ホストと'C54xによるアクセス .....          | 4-13 |
| 4-6.  | HPI-8のタイミング図 .....                         | 4-15 |
| 4-7.  | DIVクロック・モードに切り替えた場合のHPI-8転送 .....          | 4-24 |
| 4-8.  | 'C54xがIDLE3モードの場合のHPI-8の転送 .....           | 4-27 |
| 4-9.  | 汎用I/O制御レジスタ(GPIOCR):MMRアドレス003Ch .....     | 4-30 |
| 4-10. | 汎用I/Oステータス・レジスタ(GPIOSR):MMRアドレス003Dh ..... | 4-32 |
| 4-11. | GPIOコードの例 .....                            | 4-36 |
| 5-1.  | HPIストローブおよび選択ロジック .....                    | 5-3  |
| 5-2.  | 共用モードでのHPI-16へのインターフェイス('VC5420) .....     | 5-7  |
| 5-3.  | HPICレジスタ .....                             | 5-8  |
| 5-4.  | HASを使用するHPIAライト .....                      | 5-10 |
| 5-5.  | HASを使用しないHPICリード .....                     | 5-11 |
| 5-6.  | 自動インクリメントを使用するHPIDリード .....                | 5-12 |
| 5-7.  | 自動インクリメントを使用するHPIDライト .....                | 5-13 |
| 5-8.  | 分離モードでのHPI-16へのインターフェイス('VC5420) .....     | 5-15 |
| 5-9.  | 分離モードでのHPIDリード .....                       | 5-16 |
| 5-10. | 分離モードでのHPIDライト .....                       | 5-17 |
| 5-11. | HPIに対する'VC5420のメモリー・マップ .....              | 5-18 |
| 5-12. | HPIとDMAの相互作用 .....                         | 5-19 |
| 5-13. | PLLからDIVへのクロック・モード変更時のHPI-16の動作 .....      | 5-22 |
| 6-1.  | 'VC5420 – 2サブシステムDSP .....                 | 6-2  |
| 6-2.  | 'VC5420のFIFO構成 .....                       | 6-3  |
| 6-3.  | FIFO動作のためのDMA構成 .....                      | 6-5  |
| 6-4.  | HPI-16と外部メモリー・インターフェイスの接続('VC5420) .....   | 6-7  |
| 6-5.  | McBSP間通信 .....                             | 6-10 |
| 7-1.  | 不連続メモリー・リードおよびI/Oリードのバス・タイミング .....        | 7-3  |
| 7-2.  | 連続メモリー・リードのバス・タイミング(n = 3 リード) .....       | 7-4  |
| 7-3.  | メモリー・ライトおよびI/Oライトのバス・タイミング .....           | 7-5  |

# 表

|       |                                                                      |      |
|-------|----------------------------------------------------------------------|------|
| 2-1.  | McBSPのインターフェイス信号 .....                                               | 2-4  |
| 2-2.  | McBSPのレジスタ .....                                                     | 2-5  |
| 2-3.  | McBSPのCPU割り込みおよびDMAイベント同期 .....                                      | 2-6  |
| 2-4.  | シリアル・ポート制御レジスタ1(SPCR1)のビット/フィールドの説明 .....                            | 2-7  |
| 2-5.  | シリアル・ポート制御レジスタ2(SPCR2)のビット/フィールドの説明 .....                            | 2-10 |
| 2-6.  | ピン制御レジスタ(PCR)のビット/フィールドの説明 .....                                     | 2-12 |
| 2-7.  | 受信制御レジスタ1(RCR1)のビット/フィールドの説明 .....                                   | 2-16 |
| 2-8.  | 受信制御レジスタ2(RCR2)のビット/フィールドの説明 .....                                   | 2-17 |
| 2-9.  | 送信制御レジスタ1(XCR1)のビット/フィールドの説明 .....                                   | 2-19 |
| 2-10. | 送信制御レジスタ2(XCR2)のビット/フィールドの説明 .....                                   | 2-20 |
| 2-11. | McBSPピンのリセット状態 .....                                                 | 2-23 |
| 2-12. | 1フレーム当たりのワード数と1ワード当たりのビット数を制御する<br>RCR[1,2]/XCR[1,2]のビット/フィールド ..... | 2-31 |
| 2-13. | McBSP受信/送信フレーム長(1、2)の設定 .....                                        | 2-31 |
| 2-14. | McBSP受信/送信ワード長の設定 .....                                              | 2-32 |
| 2-15. | 12ビットのデータ0xABCでのRJJUSTフィールドの使用 .....                                 | 2-52 |
| 2-16. | 20ビットのデータ0xABCDEでのRJJUSTフィールドの使用 .....                               | 2-52 |
| 2-17. | サンプル・レート生成器レジスタ1(SRGR1)のビット/フィールドの説明 .....                           | 2-59 |
| 2-18. | サンプル・レート生成器レジスタ2(SRGR2)のビット/フィールドの説明 .....                           | 2-60 |
| 2-19. | 受信クロックの選択 .....                                                      | 2-65 |
| 2-20. | 送信クロックの選択 .....                                                      | 2-65 |
| 2-21. | 受信フレーム同期の選択 .....                                                    | 2-67 |
| 2-22. | 送信フレーム同期の選択 .....                                                    | 2-68 |
| 2-23. | マルチチャネル制御レジスタ1(MCR1)のビット/フィールドの説明 .....                              | 2-73 |
| 2-24. | マルチチャネル制御レジスタ2(MCR2)のビット/フィールドの説明 .....                              | 2-75 |
| 2-25. | 受信チャネル・イネーブル・レジスタ・パーティションA(RCERA)の<br>ビット/フィールドの説明 .....             | 2-80 |
| 2-26. | 受信チャネル・イネーブル・レジスタ・パーティションB(RCERB)の<br>ビット/フィールドの説明 .....             | 2-81 |
| 2-27. | 送信チャネル・イネーブル・レジスタ・パーティションA(XCERA)の<br>ビット/フィールドの説明 .....             | 2-81 |
| 2-28. | 送信チャネル・イネーブル・レジスタ・パーティションB(XCERB)の<br>ビット/フィールドの説明 .....             | 2-82 |
| 2-29. | クロック・ストップ・モードの設定 .....                                               | 2-88 |
| 2-30. | SPIモード設定用のレジスタ・ビット値 .....                                            | 2-90 |
| 2-31. | SPIマスタ動作用のレジスタ・ビット値 .....                                            | 2-91 |
| 2-32. | SPIスレーブ・オペレーション用のレジスタ・ビット値 .....                                     | 2-92 |

表

|       |                                                   |      |
|-------|---------------------------------------------------|------|
| 2-33. | McBSPクロック設定                                       | 2-95 |
| 2-34. | 汎用I/Oとしてのピンの設定                                    | 2-97 |
| 3-1.  | DMAレジスタ                                           | 3-6  |
| 3-2.  | DMAチャネル優先順位およびイネーブル制御(DMPREC)レジスタの<br>ビット/フィールド説明 | 3-8  |
| 3-3.  | 'C5402での多重割り込みの割り当て                               | 3-10 |
| 3-4.  | 'C5410での多重割り込みの割り当て                               | 3-10 |
| 3-5.  | 'C5420での多重割り込みの割り当て(各サブシステム)                      | 3-10 |
| 3-6.  | DMA同期イベントおよびフレーム・カウント(DMSFCn)レジスタの<br>ビット/フィールド説明 | 3-13 |
| 3-7.  | 'C5402でのDMA同期イベント                                 | 3-14 |
| 3-8.  | 'C5410でのDMA同期イベント                                 | 3-15 |
| 3-9.  | 'C5420でのDMA同期イベント(各サブシステム)                        | 3-16 |
| 3-10. | DMA転送モード制御(DMMCRn)レジスタのビット/フィールド説明                | 3-17 |
| 3-11. | ABUバッファの例                                         | 3-24 |
| 3-12. | ABUのアドレス・インデックス・モード                               | 3-25 |
| 3-13. | DMAブロック転送の割り込み発生モード                               | 3-28 |
| 3-14. | 'C5402のDMAメモリー・マップ                                | 3-34 |
| 3-15. | 'C5410のDMAメモリー・マップ                                | 3-35 |
| 3-16. | 'C5420のDMAメモリー・マップ                                | 3-37 |
| 3-17. | DMA転送サイクル・タイム                                     | 3-39 |
| 4-1.  | 拡張8ビットHPIとスタンダード8ビットHPIのおもな違い                     | 4-2  |
| 4-2.  | HPI-8レジスタの説明                                      | 4-5  |
| 4-3.  | HPI-8の各信号の名称とその機能                                 | 4-6  |
| 4-4.  | HPI-8の入力制御信号とその機能                                 | 4-9  |
| 4-5.  | HPI制御レジスタ(HPIC)のビット説明                             | 4-12 |
| 4-6.  | ホストと'C54xによるHPICのリード/ライトの特徴                       | 4-13 |
| 4-7.  | アクセス種類別のHPI-8内部遅延                                 | 4-17 |
| 4-8.  | ウェートステートの発生条件                                     | 4-18 |
| 4-9.  | BOBとHPIAの初期化                                      | 4-20 |
| 4-10. | HPI-8へのリード・アクセス(自動インクリメントあり)                      | 4-21 |
| 4-11. | HPIへのライトアクセス(自動インクリメントあり)                         | 4-22 |
| 4-12. | リセット時のHPI-8の動作('C5410のみ)                          | 4-29 |
| 4-13. | 汎用I/O制御レジスタ(GPIOCR)のビット機能                         | 4-30 |
| 4-14. | 汎用I/Oステータス・レジスタ(GPIOSR)のビット機能                     | 4-32 |
| 5-1.  | HPI-16ピンの説明                                       | 5-4  |
| 5-2.  | HCNTL0/1モード                                       | 5-8  |
| 5-3.  | HPICビットの説明                                        | 5-8  |
| 6-1.  | FIFO転送をサポートするDMA設定                                | 6-4  |
| 6-2.  | EMIF/HPI-16モード                                    | 6-8  |
| 6-3.  | リセット時のMP/MCビット・レベル                                | 6-8  |

|       |                                                               |      |
|-------|---------------------------------------------------------------|------|
| 2-1.  | 受信部動作中の送信部のリセットおよび設定 .....                                    | 2-26 |
| 3-1.  | サブアドレッシング使用例(自動インクリメントなし) .....                               | 3-5  |
| 3-2.  | サブアドレッシング使用例(自動インクリメントあり) .....                               | 3-5  |
| 3-2   | サブアドレッシング使用例(自動インクリメントあり)(続き) .....                           | 3-6  |
| 3-3.  | アドレス修飾によるデータのソート .....                                        | 3-22 |
| 3-4.  | ABUバッファ・サイズの例 .....                                           | 3-25 |
| 3-5.  | インデックス・アドレッシングでシングル・ワード転送を行う場合の<br>折り返しアドレス算出法 .....          | 3-26 |
| 3-6.  | インデックス・アドレッシングでダブル・ワード転送を行う場合の<br>折り返しアドレス算出法 .....           | 3-26 |
| 3-7.  | 'C5402/'C5420におけるABU割り込み例<br>+1インデックス使用時にバッファ・サイズが偶数の場合 ..... | 3-29 |
| 3-8.  | 'C5402/'C5420におけるABU割り込み例<br>+1インデックス使用時にバッファ・サイズが奇数の場合 ..... | 3-29 |
| 3-9.  | 'C5402/'C5420におけるABU割り込み例<br>-1インデックス使用時にバッファ・サイズが偶数の場合 ..... | 3-30 |
| 3-10. | 'C5402/'C5420におけるABU割り込み例<br>-1インデックス使用時にバッファ・サイズが奇数の場合 ..... | 3-30 |
| 3-11. | 'C5410におけるABU割り込み例<br>+1インデックス使用時にバッファ・サイズが偶数の場合 .....        | 3-31 |
| 3-12. | 'C5410におけるABU割り込み例<br>+1インデックス使用時にバッファ・サイズが奇数の場合 .....        | 3-31 |
| 3-13. | 'C5410におけるABU割り込み例<br>-1インデックス使用時にバッファ・サイズが偶数の場合 .....        | 3-32 |
| 3-14. | 'C5410におけるABU割り込み例<br>-1インデックス使用時にバッファ・サイズが奇数の場合 .....        | 3-32 |
| 3-15. | DMAチャンネル転送速度の例 .....                                          | 3-40 |
| 3-16. | プログラム・メモリーからデータ・メモリーへの転送(自動インクリメントなし) .....                   | 3-46 |
| 3-17. | プログラム・メモリーからデータ・メモリーへの転送(自動インクリメントあり) .....                   | 3-48 |
| 3-18. | プログラム・メモリーからデータ・メモリーへの転送(自動初期化あり) .....                       | 3-50 |
| 3-19. | ABUモードでのMcBSPデータ転送 .....                                      | 3-52 |
| 3-20. | ダブルワード・モードでのMcBSPデータ転送 .....                                  | 3-54 |
| 3-21. | データのソートを伴うMcBSPからデータ・メモリーへの転送 .....                           | 3-56 |
| 6-1.  | DMAチャンネル0および1のFIFO送受信の設定 .....                                | 6-4  |



## はじめに

'C54デバイスは、TMS320ファミリーの固定小数点デジタル・シグナル・プロセッサ(DSP)です。改良型ハーバード・アーキテクチャーを採用した中央処理装置(CPU)により、電力消費を最小限に抑え、高度な並列機能を提供します。また、多様なアドレッシング・モードと命令セットにより、システム性能がさらに強化されています。'C54xはこのほかにも多くの機能を備えているため、テレコミュニケーションなどの各リアルタイム組み込みアプリケーション固有のニーズを満たすことができます。

'C54xデバイスはすべて、汎用I/Oピン(XFおよび $\overline{\text{BIO}}$ )、タイマ('C5402では2個)、クロック生成器、プログラマブル・ウェイトステート発生器、およびプログラマブル・バンク・スイッチング・モジュールを備えています。シリアル・ポート、ホスト・ポート・インターフェイス、およびクロック生成器の種類と数は、各'C54xデバイスごとに異なります。

この章では、'C5402、'C5410、および'C5420デバイスで使用可能な高機能ペリフェラルについて説明します。

## Topic

## Page

|     |                         |     |
|-----|-------------------------|-----|
| 1.1 | 'C54x高機能ペリフェラルの概要 ..... | 1-2 |
|-----|-------------------------|-----|

## 1.1 'C54x高機能ペリフェラルの概要

以下の各節では、'C54xで使用可能な高機能ペリフェラルの概要を説明します。

### 1.1.1 マルチチャネル・バッファド・シリアル・ポート(McBSP)

'C54xファミリーは、システム内の他の'C54xデバイス、コーデックその他のデバイスに直接インターフェイスできる、高速に全二重双方向通信を実現するマルチチャネル・バッファド・シリアル・ポートを備えています。McBSPは、他の'C54xデバイスに使用されているスタンダード・シリアル・ポート・インターフェイスの高機能バージョンです。McBSPには、次のような特長があります。

- ☐ 連続したデータ・ストリームの扱いを可能にする、ダブル・バッファ構成の送信動作とトリプル・バッファ構成の受信動作
- ☐ 受信側と送信側とでそれぞれ独立したフレーム生成/同期とクロック生成/同期
- ☐ 最大128チャネルのマルチチャネル送受信
- ☐ 8、12、16、20、24、および32ビットのデータ・サイズ
- ☐  $\mu$ 則およびA則圧伸処理
- ☐ フレーム同期およびクロックのプログラムによる極性設定
- ☐ 内部クロックおよびフレーム生成のプログラムによる設定

McBSPの詳細については、第2章を参照してください。

### 1.1.2 ダイレクト・メモリー・アクセス(DMA:Direct Memory Access)コントローラ

6チャネルの'C54xダイレクト・メモリー・アクセス(DMA:Direct Memory Access)コントローラはメモリー・マップ上の領域間におけるデータ転送をCPUとは独立に実行します。DMAにより、内部メモリ(プログラム/データ)、McBSPなどの内部ペリフェラル、外部メモリー・デバイス間でのデータのやりとりをCPU動作のバックグラウンドで行うことができます。DMAには6つの独立したプログラマブル・チャネルがあり、これによって6つの異なるデータ転送設定が可能です。DMAの詳細については、第3章を参照してください。

### 1.1.3 ホスト・ポート・インターフェイス(HPI-8およびHPI-16)

'C54xには2つの拡張ホスト・ポート・インターフェイス、HPI-8とHPI-16があります。これらは、ホスト・プロセッサへのインターフェイスを提供する、8ビットおよび16ビットの平行・ポートです。'C54xとホスト・プロセッサの間の情報交換は、ホスト・プロセッサと'C54xの両方からアクセスできる'C54xオンチップ・メモリー上で行われます。HPI-8とHPI-16の動作の詳細については、それぞれ第4章と第5章を参照してください。

# マルチチャネル・バッファド・シリアル・ポート (McBSP)

'C54xデジタル・シグナル・プロセッサの一部のデバイスには、高速かつ多様な全二重双方向のマルチチャネル・バッファド・シリアル・ポート (McBSP) を複数搭載しています。これによって、他の'C54xデバイス、コーデック、およびシステム内の他のデバイスに直接インターフェイスできます。'C5402は2個、'C5410は3個、'C5420は6個のMcBSPを備えています。これらのMcBSPは、他の'C54xデバイスに使用されているスタンダード・シリアル・ポート・インターフェイスをベースとしています。

この章では、McBSPの動作について説明し、レジスタの定義とタイミング図を示します。

| Topic                                                | Page |
|------------------------------------------------------|------|
| 2.1 McBSPの特長 .....                                   | 2-2  |
| 2.2 McBSPの概要 .....                                   | 2-3  |
| 2.3 データ送受信フロー .....                                  | 2-22 |
| 2.4 $\mu$ -law/A-law圧伸ハードウェアの動作: (R/X) COMPAND ..... | 2-53 |
| 2.5 プログラマブル・クロックおよびフレーム生成 .....                      | 2-57 |
| 2.6 マルチチャネル選択動作 .....                                | 2-72 |
| 2.7 SPIプロトコル: McBSPクロック・ストップ・モード .....               | 2-86 |
| 2.8 エミュレーションFREEビットおよびSOFTビット .....                  | 2-95 |
| 2.9 McBSPの汎用入出力ポート .....                             | 2-96 |
| 2.10 パワーダウン・モードにおけるMcBSPの動作 .....                    | 2-98 |
| 2.11 McBSPプログラミングのサンプル・コード .....                     | 2-99 |

## 2.1 McBSPの特長

McBSPは、TMS320C2x、'C20x、'C5x、および'C54xデバイスに使用されているスタンダード・シリアル・ポート・インターフェイスをベースとしています。McBSPは次の機能を提供します。

- ☐ 全二重双方向通信
  - ☐ 連続したデータ・ストリームの扱いを可能にする、ダブル・バッファ構成の送信データ・レジスタとトリプル・バッファ構成の受信データ・レジスタ
  - ☐ 送受信のための独立したフレーム生成/同期およびクロック生成/同期
  - ☐ 業界標準コーデック、アナログ・インターフェイス・チップ(AIC)、ならびに他のシリアル接続A/DデバイスおよびD/Aデバイスへの直接インターフェイス
  - ☐ 外部シフト・クロック生成、または周波数のプログラムが可能な内部シフト・クロック
- さらに、McBSPは次の機能を備えています。
- ☐ 以下のデバイスへの直接インターフェイス
    - T1/E1フレーム
    - 以下のMVIPスイッチング互換およびST-BUS準拠のデバイス
      - MVIPフレーム
      - H.100フレーム
      - SCSAフレーム
    - IOM-2準拠のデバイス
    - AC97準拠のデバイス(マルチフェーズ・フレーム同期機能が備えられています。)
    - IIS準拠のデバイス
    - SPI™ デバイス
  - ☐ 最大128チャンネルのマルチチャンネル送受信
  - ☐ 8、12、16、20、24、および32ビットの多様なデータ・サイズ

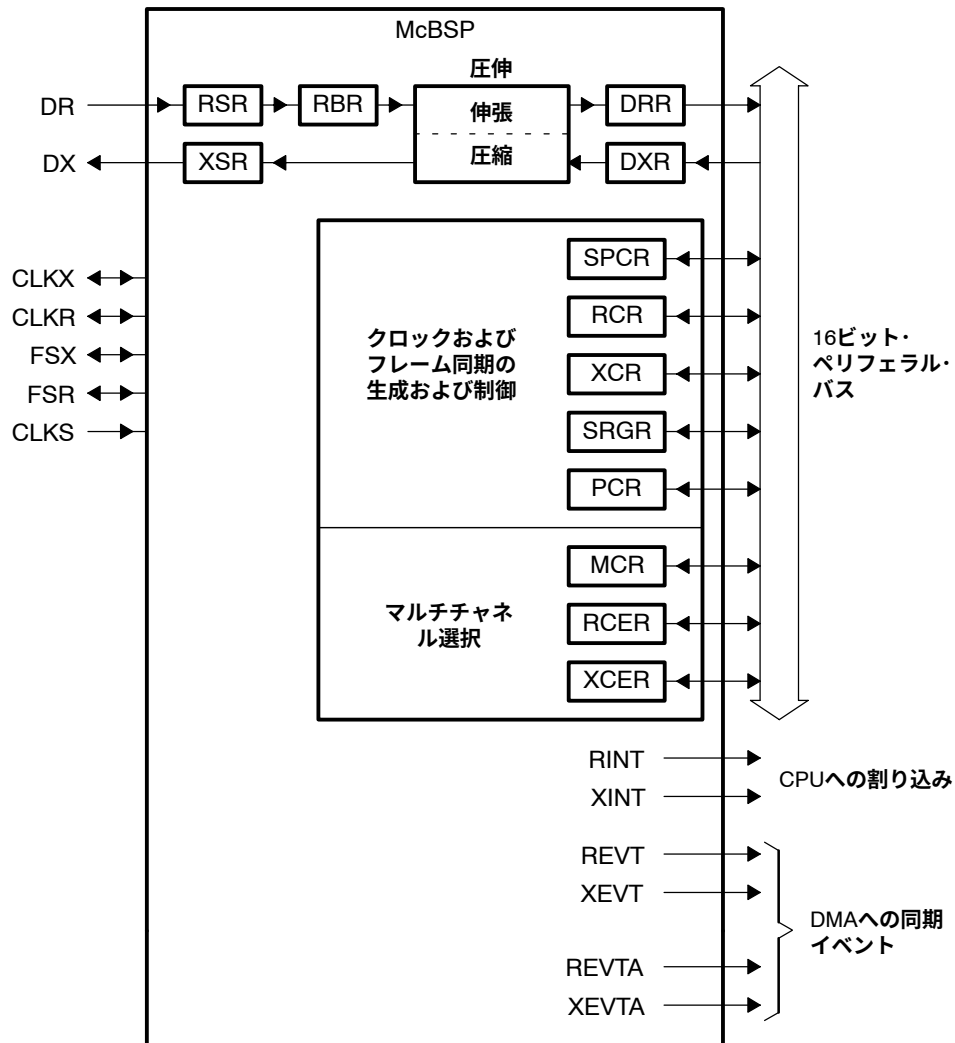
注記: 本書では、データ・サイズはワードまたはシリアル・ワードとして表記され、8、12、16、20、24、または32ビットのいずれかになります。これは、ワード長を16ビットとするワードの真の定義とは異なります。
  - ☐  $\mu$ 則およびA則圧伸
  - ☐ LSBからまたはMSBからというオプション付きの8ビット・データ伝送
  - ☐ フレーム同期およびデータ・クロックの両方の極性をプログラマブルに設定可能
  - ☐ 内部クロックおよびフレーム生成のプログラムによる柔軟な設定

## 2.2 McBSPの概要

McBSPは、図2-1のように、データ・バスと制御バスの合計7本のピンによって外部デバイスと接続されます。

2

図2-1. McBSPの内部ブロック図



McBSPとインターフェイスをとるデバイスへのデータ送受信は、送信についてはデータ送信ピン(DX)、受信についてはデータ受信ピン(DR)を経由して行われます。制御信号(クロックとフレーム同期)は、CLKX、CLKR、FSX、およびFSR経由で送受信されます。C54xは、内部ペリフェラル・バス経由でアクセス可能な16ビット幅の制御レジスタと通信します。

CPUまたはDMAコントローラは、受信したデータをデータ受信レジスタ(DRR[1,2])からリードして、送信するデータをデータ送信レジスタ(DXR[1,2])にライトします。DXR[1,2]にライトされたデータは、送信シフト・レジスタ(XSR[1,2])経由でDXピンにシフト・アウトされます。同様に、DRピン上の受信データは受信シフト・レジスタ(RSR[1,2])にシフト・インされ、受信バッファ・レジスタ(RBR[1,2])内にコピーされます。次に、RBR[1,2]がDRR[1,2]にコピーされます。CPUまたはDMAコントローラはDRR[1,2]をリードすることができます。これによって、内部および外部のデータ通信を同時に行うことができます。

DRR2、RBR2、RSR2、DXR2、およびXSR2の各レジスタは、受信/送信ワード長(R/XWDLEN[1,2])が8ビット、12ビット、または16ビット・モードに設定されている場合は使用されません(ライト、リード、またはシフトが行われません)。

CPUからアクセス可能な残りのレジスタは、McBSPの制御方法を設定します。2-5ページの表2-2「McBSPのレジスタ」に、それらのレジスタを示します。制御ブロックは、内部クロック生成、フレーム同期信号生成、およびそれらの制御とマルチチャネル選択で構成されます。この制御ブロックは、2-6ページの表2-3「McBSPのCPU割り込みおよびDMAイベント同期」に示す2つの割り込みと4つのイベント信号により、重要なイベントの通知をCPUおよびDMAコントローラに送信します。

表2-1. McBSPのインターフェイス信号

| ピン   | I/O/Z <sup>†</sup> | 説明         |
|------|--------------------|------------|
| CLKR | I/O/Z              | 受信クロック     |
| CLKX | I/O/Z              | 送信クロック     |
| CLKS | I                  | 外部クロック     |
| DR   | I                  | 受信シリアル・データ |
| DX   | O/Z                | 送信シリアル・データ |
| FSR  | I/O/Z              | 受信フレーム同期   |
| FSX  | I/O/Z              | 送信フレーム同期   |

<sup>†</sup> I=入力、O=出力、Z=ハイ・インピーダンス

表2-2. McBSPのレジスタ

| 16進アドレス |         |         | サブアドレ<br>ス | 名称       | レジスタ名 <sup>†</sup>                      | 参照節     |
|---------|---------|---------|------------|----------|-----------------------------------------|---------|
| McBSP 0 | McBSP 1 | McBSP 2 |            |          |                                         |         |
| —       | —       | —       |            | RBR[1,2] | McBSP受信バッファ・レジスタ1および2                   | 2.2     |
| —       | —       | —       |            | RSR[1,2] | McBSP受信シフト・レジスタ1および2                    | 2.2     |
| —       | —       | —       |            | XSR[1,2] | McBSP送信シフト・レジスタ1および2                    | 2.2     |
| 0020    | 0040    | 0030    | —          | DRR2x    | McBSPデータ受信レジスタ2                         | 2.2     |
| 0021    | 0041    | 0031    | —          | DRR1x    | McBSPデータ受信レジスタ1                         | 2.2     |
| 0022    | 0042    | 0032    | —          | DXR2x    | McBSPデータ送信レジスタ2                         | 2.2     |
| 0023    | 0043    | 0033    | —          | DXR1x    | McBSPデータ送信レジスタ1                         | 2.2     |
| 0038    | 0048    | 0034    | —          | SPSAx    | McBSPサブアドレス・レジスタ                        |         |
| 0039    | 0049    | 0035    | 0x0000     | SPCR1x   | McBSPシリアル・ポート制御レジスタ1                    | 2.2.1   |
| 0039    | 0049    | 0035    | 0x0001     | SPCR2x   | McBSPシリアル・ポート制御レジスタ2                    | 2.2.1   |
| 0039    | 0049    | 0035    | 0x0002     | RCR1x    | McBSP受信制御レジスタ1                          | 2.2.2   |
| 0039    | 0049    | 0035    | 0x0003     | RCR2x    | McBSP受信制御レジスタ2                          | 2.2.2   |
| 0039    | 0049    | 0035    | 0x0004     | XCR1x    | McBSP送信制御レジスタ1                          | 2.2.2   |
| 0039    | 0049    | 0035    | 0x0005     | XCR2x    | McBSP送信制御レジスタ2                          | 2.2.2   |
| 0039    | 0049    | 0035    | 0x0006     | SRGR1x   | McBSPサンプル・レート生成器レジスタ1                   | 2.5.1.1 |
| 0039    | 0049    | 0035    | 0x0007     | SRGR2x   | McBSPサンプル・レート生成器レジスタ2                   | 2.5.1.1 |
| 0039    | 0049    | 0035    | 0x0008     | MCR1x    | McBSPマルチチャネル・レジスタ1                      | 2.6.1   |
| 0039    | 0049    | 0035    | 0x0009     | MCR2x    | McBSPマルチチャネル・レジスタ2                      | 2.6.1   |
| 0039    | 0049    | 0035    | 0x000A     | RCERAx   | RCERAx McBSP受信チャンネル・イネーブル・レジスタ・パーティションA | 2.6.3.1 |
| 0039    | 0049    | 0035    | 0x000B     | RCERBx   | McBSP受信チャンネル・イネーブル・レジスタ・パーティションB        | 2.6.3.1 |

<sup>†</sup> RBR[1,2]、RSR[1,2]、およびXSR[1,2]は、CPUまたはDMAから直接アクセスできません。

表2-2. McBSPのレジスタ(続き)

| 16進アドレス |         |         | サブアドレ<br>ス | 名称     | レジスタ名 <sup>†</sup>              | 参照節             |
|---------|---------|---------|------------|--------|---------------------------------|-----------------|
| McBSP 0 | McBSP 1 | McBSP 2 |            |        |                                 |                 |
| 0039    | 0049    | 0035    | 0x000C     | XCERAx | McBSP送信チャネル・イネーブル・レジスタ・パーティションA | 2.6.3.1         |
| 0039    | 0049    | 0035    | 0x000D     | XCERBx | McBSP送信チャネル・イネーブル・レジスタ・パーティションB | 2.6.3.1         |
| 0039    | 0049    | 0035    | 0x000E     | PCRx   | McBSPピン制御レジスタ                   | 2.2.1<br>および2.9 |

<sup>†</sup> RBR[1,2]、RSR[1,2]、およびXSR[1,2]は、CPUまたはDMAから直接アクセスできません。

表2-3. McBSPのCPU割り込みおよびDMAイベント同期

| 割り込み名 | 説明             | 節       |
|-------|----------------|---------|
| RINT  | CPUへの受信割り込み    | 2.3.3   |
| XINT  | CPUへの送信割り込み    | 2.3.3   |
| REVT  | DMAへの受信同期イベント  | 2.3.2.1 |
| XEVT  | DMAへの送信同期イベント  | 2.3.2.2 |
| REVTa | DMAへの受信同期イベントA | 2.6.4   |
| XEVTa | DMAへの送信同期イベントA | 2.6.4   |

### 2.2.1 シリアル・ポートの設定

シリアル・ポートは、2個の16ビットのシリアル・ポート制御レジスタ1および2(SPCR[1,2])とピン制御レジスタ(PCR)によって設定されます。図2-2、図2-3、および図2-4に各レジスタの内容を示します。SPCR[1,2]とPCRには、McBSPのステータス情報とともに、希望の動作を設定できるビットが含まれています。各ビット/フィールドの動作については、2-7ページの表2-4「シリアル・ポート制御レジスタ1(SPCR1)のビット/フィールドの説明」、2-10ページの表2-5「シリアル・ポート制御レジスタ2(SPCR2)のビット/フィールドの説明」、および2-12ページの表2-6「ピン制御レジスタ(PCR)のビット/フィールドの説明」の中で説明します。



PCRは、McBSPピンを通常のシリアル・ポート動作中の入出力として設定するためだけでなく、受信部または送信部がリセット時にシリアル・ポート・ピンを汎用入力または汎用出力として設定するためにも使用されます。これについては、2-96ページの第2.9節「McBSPの汎用入出力ポート」で説明します。

図2-2. シリアル・ポート制御レジスタ(SPCR1)

|       |       |       |        |          |       |                         |
|-------|-------|-------|--------|----------|-------|-------------------------|
| 15    | 14    | 13    | 12     | 11       | 10    | 8                       |
| DLB   | RJUST |       | CLKSTP |          | 予約    |                         |
| RW,+0 | RW,+0 | RW,+0 | RW,+0  |          | R,+0  |                         |
| 7     | 6     | 5     | 4      | 3        | 2     | 1 0                     |
| DXENA | ABIS  | RINTM |        | RSYNCERR | RFULL | RRDY Rrst               |
| RW,+0 | RW,+0 | RW,+0 |        | RW,+0    | R,+0  | R,+0 RW,+0 <sup>†</sup> |

注: R = リード、W = ライト、+0 = リセット時の値

<sup>†</sup> \* R, +0は、リード専用でリセット値が0であるという意味です。RW, +0は、リードとライトが可能でリセット値が0であるという意味です。

表2-4. シリアル・ポート制御レジスタ1(SPCR1)のビット/フィールドの説明

| ビット   | 名前    | 機能                                                        | 節       |
|-------|-------|-----------------------------------------------------------|---------|
| 15    | DLB   | デジタル・ループ・バック・モード                                          | 2.5.2.5 |
|       |       | DLB = 0                      デジタル・ループ・バック・モードのディセーブル      |         |
|       |       | DLB = 1                      デジタル・ループ・バック・モードのイネーブル       |         |
| 14-13 | RJUST | 受信データの符号拡張および格納方法                                         | 2.3.8   |
|       |       | RJUST = 00                      右詰めおよびDRR[1,2]の上位ビットをゼロ詰め |         |
|       |       | RJUST = 01                      右詰めおよびDRR[1,2]の上位ビットを符号拡張 |         |
|       |       | RJUST = 10                      左詰めおよびDRR[1,2]の下位ビットをゼロ詰め |         |
|       |       | RJUST = 11                      予約                        |         |

表2-4. シリアル・ポート制御レジスタ1(SPCR1)のビット/フィールドの説明 (続き)

| ビット   | 名前       | 機能                                                                                                                                                                                                                                                                                                                                                                    | 節       |
|-------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| 12-11 | CLKSTP   | <b>クロック・ストップ・モード</b><br><br>CLKSTP = 0X      クロック・ストップ・モードのディセーブル。非SPIモード用の通常のクロック同期<br><br><b>次の場合は各種SPIモード:</b><br><br>CLKSTP = 10 および CLKXP = 0      クロックが遅延なしの立ち上がりエッジで始動<br><br>CLKSTP = 10 および CLKXP = 1      クロックが遅延なしの立ち下がりエッジで始動<br><br>CLKSTP = 11 および CLKXP = 0      クロックが遅延をとまなう立ち上がりエッジで始動<br><br>CLKSTP = 11 および CLKXP = 1      クロックが遅延をとまなう立ち下がりエッジで始動 | 2.7     |
| 10-8  | 予約       | 予約                                                                                                                                                                                                                                                                                                                                                                    |         |
| 7     | DXENA    | <b>DXイネーブラ</b><br><br>DXENA = 0      DXイネーブラのオフ<br><br>DXENA = 1      DXイネーブラのオン                                                                                                                                                                                                                                                                                      | 2.3.4.8 |
| 6     | ABIS     | <b>ABISモード</b><br><br>ABIS = 0      A-bisモードのディセーブル<br><br>ABIS = 1      A-bisモードのイネーブル                                                                                                                                                                                                                                                                               | 2.6.4   |
| 5-4   | RINTM    | <b>受信割り込みモード</b><br><br>RINTM = 00      RINTがRRDY(ワード終端)およびA-bisモードのフレームの終わりで生成されます。<br><br>RINTM = 01      RINTがマルチチャネル動作のブロック終端またはフレームの終わりで生成されます。<br><br>RINTM = 10      RINTが新規フレーム同期によって生成されます。<br><br>RINTM = 11      RINTがRSYNCERRによって生成されます。                                                                                                                  | 2.3.3   |
| 3     | RSYNCERR | <b>受信同期エラー</b><br><br>RSYNCERR = 0      同期エラーなし<br><br>RSYNCERR = 1      McBSPが同期エラーを検出                                                                                                                                                                                                                                                                               | 2.3.7.2 |

表2-4. シリアル・ポート制御レジスタ1(SPCR1)のビット/フィールドの説明 (続き)

| ビット | 名前                       | 機能                                                                                                                                                                                      | 節       |
|-----|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| 2   | RFULL                    | 受信シフト・レジスタ(RSR[1,2])がフル<br><br>RFULL = 0                      RBR[1,2]がオーバーラン状態ではありません。<br><br>RFULL = 1                      DRR[1,2]がリードされず、RBR[1,2]がフルで、<br>RSR[1,2]も新規受信ワードでフル状態です。 | 2.3.7.1 |
| 1   | RRDY                     | 受信部レディー<br><br>RRDY = 0                      受信部がレディー状態ではありません。<br><br>RRDY = 1                      DRR[1,2]からのデータ・リードについて受信部がレ<br>ディー状態です。                                            | 2.3.2   |
| 0   | $\overline{\text{RRST}}$ | 受信部リセット。受信部をリセット及びイネーブルにします。<br><br>$\overline{\text{RRST}} = 0$ シリアル・ポート受信部をディセーブルかつリセッ<br>ト状態<br><br>$\overline{\text{RRST}} = 1$ シリアル・ポート受信部をイネーブルにします。                              | 2.3.1   |

図2-3. シリアル・ポート制御レジスタ2(SPCR2)

|                 |       |       |    |                       |        |       |       |
|-----------------|-------|-------|----|-----------------------|--------|-------|-------|
| 15              | 14    | 13    | 12 | 11                    | 10     | 9     | 8     |
| 予約 <sup>†</sup> |       |       |    |                       |        | FREE  | SOFT  |
| R,+0            |       |       |    |                       |        | RW,+0 | RW,+0 |
| 7               | 6     | 5     | 4  | 3                     | 2      | 1     | 0     |
| FRST            | GRST  | XINTM |    | XSUNCERR <sup>‡</sup> | XEMPTY | XRDY  | XRST  |
| RW,+0           | RW,+0 | RW,+0 |    | RW,+0                 | R,+0   | R,+0  | RW,+0 |

注: R = リード、W = ライト、+0 = リセット時の値

<sup>†</sup> 注記: このビット/フィールドとすべての予約ビット/フィールドには、対応する記憶領域は設定されていません。しかし、これらのビット/フィールドは常に0として読み出されます。

<sup>‡</sup> 注意: このビットに1をライトすると、エラー状態が設定されます。したがってこのビットへのライトは主にテストに使用されるか、この動作が望まれる場合に使用されます。

表2-5. シリアル・ポート制御レジスタ2(SPCR2)のビット/フィールドの説明

| ビット   | 名前   | 機能                                                                                          | 節     |
|-------|------|---------------------------------------------------------------------------------------------|-------|
| 15-10 | 予約   | 予約                                                                                          |       |
| 9     | FREE | フリー・ランニング・モード                                                                               | 2.8   |
|       |      | FREE = 0      フリー・ランニング・モードのディセーブル                                                          |       |
|       |      | FREE = 1      フリー・ランニング・モードのイネーブル                                                           |       |
| 8     | SOFT | ソフト・ビット                                                                                     | 2.8   |
|       |      | SOFT = 0      SOFTモードのディセーブル                                                                |       |
|       |      | SOFT = 1      SOFTモードのイネーブル                                                                 |       |
| 7     | FRST | フレーム同期生成器リセット                                                                               | 2.3.1 |
|       |      | FRST = 0      フレーム同期ロジックがリセットされます。サンプル・レート生成器によってフレーム同期信号FSGが生成されません。                       |       |
|       |      | FRST = 1      (FPER+1)回のCLKGクロックをカウント後、フレーム同期信号FSGが生成されます。すべてのフレーム・カウンタにプログラムされた値がロードされます。  |       |
| 6     | GRST | サンプル・レート生成器リセット                                                                             | 2.3.1 |
|       |      | GRST = 0      サンプル・レート生成器がリセットされます。                                                         |       |
|       |      | GRST = 1      サンプル・レート生成器のリセットが解除されます。CLKGは、サンプル・レート生成器レジスタ(SRGR[1,2])のプログラムされた値によって生成されます。 |       |

表2-5. シリアル・ポート制御レジスタ2(SPCR2)のビット/フィールドの説明 (続き)

| ビット | 名前                         | 機能                                                               | 節       |
|-----|----------------------------|------------------------------------------------------------------|---------|
| 5-4 | XINTM                      | 送信割り込みモード                                                        | 2.3.3   |
|     |                            | XINTM = 00      XINTがXRDY(ワード終端)およびA-bisモードのフレームの終わりで生成されます。     |         |
|     |                            | XINTM = 01      XINTがマルチチャネル動作でブロック終端またはフレームの終わりで生成されます。         |         |
|     |                            | XINTM = 10      XINTが新規フレーム同期によって生成されます。                         |         |
|     |                            | XINTM = 11      XINTがXSYNCERRによって生成されます。                         |         |
| 3   | XSYNCERR                   | 送信同期エラー                                                          | 2.3.7.2 |
|     |                            | XSYNCERR = 0      同期エラーなし                                        |         |
|     |                            | XSYNCERR = 1      McBSPが同期エラーを検出                                 |         |
| 2   | $\overline{\text{XEMPTY}}$ | 送信シフト・レジスタ(XSR[1,2])エンプティ                                        | 2.3.7.4 |
|     |                            | $\overline{\text{XEMPTY}}$ = 0      XSR[1,2]が空                   |         |
|     |                            | $\overline{\text{XEMPTY}}$ = 1      XSR[1,2]が空ではない               |         |
| 1   | XRDY                       | 送信部レディー                                                          | 2.3.2   |
|     |                            | XRDY = 0      送信部がレディー状態ではありません                                  |         |
|     |                            | XRDY = 1      DXR[1,2]内の新規データについて送信部がレディー状態                      |         |
| 0   | $\overline{\text{XRST}}$   | 送信部・リセット                                                         | 2.3.1   |
|     |                            | $\overline{\text{XRST}}$ = 0      シリアル・ポート送信部をディセーブルかつリセット状態にします |         |
|     |                            | $\overline{\text{XRST}}$ = 1      シリアル・ポート送信部をイネーブルにします          |         |

図2-4. ピン制御レジスタ(PCR)

|      |           |         |         |       |       |       |       |
|------|-----------|---------|---------|-------|-------|-------|-------|
| 15   | 14        | 13      | 12      | 11    | 10    | 9     | 8     |
| 予約   | XIOEN     | RIOEN   | FSXM    | FSRM  | CLKXM | CLKRM |       |
| R,+0 | RW,+0     | RW,+0   | RW,+0   | RW,+0 | RW,+0 | RW,+0 |       |
| 7    | 6         | 5       | 4       | 3     | 2     | 1     | 0     |
| 予約   | CLKS_STAT | DX_STAT | DR_STAT | FSXP  | FSRP  | CLKXP | CLKRP |
| R,+0 | R,+0      | R,+0    | R,+0    | RW,+0 | RW,+0 | RW,+0 | RW,+0 |

注: R = リード、W = ライト、+0 = リセット時の値

表2-6. ピン制御レジスタ(PCR)のビット/フィールドの説明

| ビット   | 名前    | 機能                                                                                                                                                                                                                                                                                  | 節   |
|-------|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 15-14 | 予約    | 予約                                                                                                                                                                                                                                                                                  |     |
| 13    | XIOEN | SPCR[1,2]の $\overline{XRST} = 0$ の場合にのみ送信汎用I/Oモード<br><br>XIOEN = 0      DX、FSX、およびCLKXがシリアル・ポート用のピンとして設定され、汎用I/Oとして機能しません。<br><br>XIOEN = 1      DXピンは汎用出力になります。FSXとCLKXは汎用入出力になります。これらのシリアル・ポート・ピンはシリアル・ポート動作を実行しません。                                                              | 2.9 |
| 12    | RIOEN | SPCR[1,2]の $\overline{RRST} = 0$ の場合にのみ受信汎用I/Oモード<br><br>RIOEN = 0      DR、FSR、CLKR、およびCLKSがシリアル・ポート用のピンとして設定され、汎用I/Oとして機能しません。<br><br>RIOEN = 1      DRピンとCLKSピンは汎用入力です。これらのシリアル・ポート・ピンはシリアル・ポート動作を行いません。FSRとCLKRは汎用入出力になります。CLKSピンは、受信部の $\overline{RRST}$ とRIOEN信号の組み合わせに影響されます。 | 2.9 |

表2-6. ピン制御レジスタ(PCR)のビット/フィールドの説明 (続き)

| ビット | 名前    | 機能                                                                                               | 節                  |
|-----|-------|--------------------------------------------------------------------------------------------------|--------------------|
| 11  | FSXM  | 送信フレーム同期モード                                                                                      | 2.5.3.3<br>および 2.9 |
|     |       | FSXM = 0      フレーム同期信号が外部ソースから供給されます。                                                            |                    |
|     |       | FSXM = 1      フレーム同期信号が、SRGR2のFSGM(サンプル・レート生成器フレーム同期ビット)によって決まります。                               |                    |
| 10  | FSRM  | 受信フレーム同期モード                                                                                      | 2.5.3.2<br>および 2.9 |
|     |       | FSRM = 0      フレーム同期信号が外部デバイスによって生成されます。FSRは入力ピンになります。                                           |                    |
|     |       | FSRM = 1      フレーム同期信号がサンプル・レート生成器によって内部で生成されます。FSRは、SRGR2のGSYNC=1の場合を除いて出力ピンになります(第2.5.1.1節参照)。 |                    |
| 9   | CLKXM | 送信クロック・モード                                                                                       | 2.5.2.7<br>および 2.9 |
|     |       | CLKXM = 0      CLKXを入力ピンとして、送信クロックが外部クロックによって生成されます。                                             |                    |
|     |       | CLKXM = 1      CLKXは出力ピンであり、内部のサンプル・レート生成器によって駆動されます。                                            |                    |
|     |       | SPIモードのとき(CLKSTPが0以外の値のとき):                                                                      |                    |
|     |       | CLKXM = 0      McBSPはスレーブであり、クロック(CLKX)はシステム内のSPIマスタによって生成されます。CLKRはCLKXによって内部で駆動されます。           |                    |
|     |       | CLKXM = 1      McBSPはマスタであり、その受信クロック(CLKR)を生成するクロック(CLKX)と、システム内のSPI対応スレーブのシフト・クロックを生成します。       |                    |

表2-6. ピン制御レジスタ(PCR)のビット/フィールドの説明 (続き)

| ビット | 名前        | 機能                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 節                 |
|-----|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| 8   | CLKRM     | <p><b>受信クロック・モード</b></p> <p>ケース1: SPCR1でデジタル・ループバック・モードが設定されていないとき (DLB = 0)</p> <p>CLKRM = 0      受信クロック(CLKR)は入力であり、外部クロックによってドライブされます。</p> <p>CLKRM = 1      CLKRは出力ピンであり、内部のサンプル・レート生成器によってドライブされます。</p> <p>ケース2: SPCR1でデジタル・ループバック・モードが設定されているとき (DLB = 1)</p> <p>CLKRM = 0      受信クロック (CLKRピンのことではない)は、PCRのCLKXMビットに基づき送信クロック(CLKX)によって生成されます。CLKRはハイ・インピーダンス状態になります。</p> <p>CLKRM = 1      CLKRは出力ピンであり、送信クロックによってドライブされます。送信クロックはPCRのCLKXMビットに基づいて決定されます。</p> | 2.5.2.6<br>および2.9 |
| 7   | 予約        | 予約                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                   |
| 6   | CLKS_STAT | CLKSピンのステータス。CLKSピンが汎用入力として選択された場合にその値を反映します。                                                                                                                                                                                                                                                                                                                                                                                                                          | 2.9               |
| 5   | DX_STAT   | DXピンのステータス。DXピンが汎用出力として選択された場合にその値を反映します。                                                                                                                                                                                                                                                                                                                                                                                                                              | 2.9               |
| 4   | DR_STAT   | DRピンのステータス。DRピンが汎用入力として選択された場合にその値を反映します。                                                                                                                                                                                                                                                                                                                                                                                                                              |                   |
| 3   | FSXP      | 送信フレーム同期極性                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 2.3.4.1<br>および2.9 |
|     |           | FSXP = 0      フレーム同期パルスFSXがアクティブ・ハイ                                                                                                                                                                                                                                                                                                                                                                                                                                    |                   |
|     |           | FSXP = 1      フレーム同期パルスFSXがアクティブ・ロー                                                                                                                                                                                                                                                                                                                                                                                                                                    |                   |
| 2   | FSRP      | 受信フレーム同期極性                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 2.3.4.1<br>および2.9 |
|     |           | FSRP = 0      フレーム同期パルスFSRがアクティブ・ハイ                                                                                                                                                                                                                                                                                                                                                                                                                                    |                   |
|     |           | FSRP = 1      フレーム同期パルスFSRがアクティブ・ロー                                                                                                                                                                                                                                                                                                                                                                                                                                    |                   |



表2-6. ピン制御レジスタ(PCR)のビット/フィールドの説明 (続き)

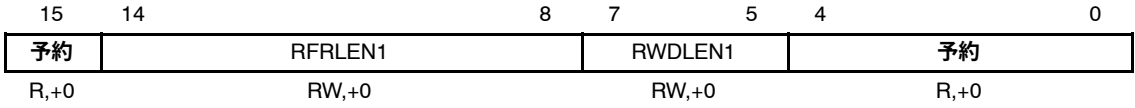
| ビット | 名前    | 機能                                             | 節                  |
|-----|-------|------------------------------------------------|--------------------|
| 1   | CLKXP | 送信クロック極性                                       | 2.3.4.1<br>および 2.9 |
|     |       | CLKXP = 0      送信データがCLKXの立ち上がりエッジでサンプリングされます。 |                    |
|     |       | CLKXP = 1      送信データがCLKXの立ち下がりエッジでサンプリングされます。 |                    |
| 0   | CLKRP | 受信クロック極性                                       | 2.3.4.1<br>および 2.9 |
|     |       | CLKRP = 0      受信データがCLKRの立ち下がりエッジでサンプリングされます。 |                    |
|     |       | CLKRP = 1      受信データがCLKRの立ち上がりエッジでサンプリングされます。 |                    |

### 2.2.2 受信制御レジスタおよび送信制御レジスタ: RCR[1,2]およびXCR[1,2]

受信制御レジスタと送信制御レジスタ(RCR[1,2]およびXCR[1,2])は、それぞれ受信動作と送信動作の各種パラメータを設定します。2-16ページの図2-5「受信制御レジスタ1(RCR1)」、2-17ページの図2-6「受信制御レジスタ2(RCR2)」、2-19ページの図2-7「送信制御レジスタ1(XCR1)」、および2-20ページの図2-8「送信制御レジスタ2(XCR2)」にこれらに示します。

各ビット/フィールドの動作については、2-16ページの表2-7「受信制御レジスタ1(RCR1)のビット/フィールドの説明」、2-17ページの表2-8「受信制御レジスタ2(RCR2)のビット/フィールドの説明」、2-19ページの表2-9「送信制御レジスタ1(XCR1)のビット/フィールドの説明」、2-20ページの表2-10「送信制御レジスタ2(XCR2)のビット/フィールドの説明」で説明します。

図2-5. 受信制御レジスタ(RCR1)



注: R = リード、W = ライト、+0 = リセット時の値

表2-7. 受信制御レジスタ1(RCR1)のビット/フィールドの説明

| ビット  | 名前      | 機能                            | 節       |
|------|---------|-------------------------------|---------|
| 15   | 予約      | 予約                            |         |
| 14-8 | RFRLN1  | 受信フレーム長1                      | 2.3.4.3 |
|      |         | RFRLN1 = 000 0000 1ワード/フレーム   |         |
|      |         | RFRLN1 = 000 0001 2ワード/フレーム   |         |
|      |         |                               |         |
|      |         |                               |         |
|      |         | RFRLN1 = 111 1111 128ワード/フレーム |         |
| 7-5  | RWDLEN1 | 受信ワード長 1                      | 2.3.4.4 |
|      |         | RWDLEN1 = 000 8ビット            |         |
|      |         | RWDLEN1 = 001 12ビット           |         |
|      |         | RWDLEN1 = 010 16ビット           |         |
|      |         | RWDLEN1 = 011 20ビット           |         |
|      |         | RWDLEN1 = 100 24ビット           |         |
|      |         | RWDLEN1 = 101 32ビット           |         |
|      |         | RWDLEN1 = 11X 予約              |         |
| 4-0  | 予約      | 予約                            |         |

図2-6. 受信制御レジスタ2(RCR2)

|        |        |         |          |       |         |   |   |   |   |
|--------|--------|---------|----------|-------|---------|---|---|---|---|
| 15     | 14     | 8       | 7        | 5     | 4       | 3 | 2 | 1 | 0 |
| RPHASE | RFLEN2 | RWDLEN2 | RCOMPAND | RFIG  | RDATDLY |   |   |   |   |
| RW,+0  | RW,+0  | RW,+0   | RW,+0    | RW,+0 | RW,+0   |   |   |   |   |

注: R = リード、W = ライト、+0 = リセット時の値

2

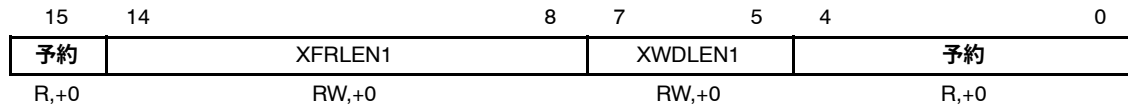
表2-8. 受信制御レジスタ2(RCR2)のビット/フィールドの説明

| ビット  | 名前      | 機能                                 | 節       |
|------|---------|------------------------------------|---------|
| 15   | RPHASE  | 受信フェーズ                             | 2.3.4.2 |
|      |         | RPHASE = 0      シングル・フェーズのフレーム     |         |
|      |         | RPHASE = 1      デュアル・フェーズのフレーム     |         |
| 14-8 | RFLEN2  | 受信フレーム長2                           | 2.3.4.3 |
|      |         | RFLEN2 = 000 0000      1ワード/フレーム   |         |
|      |         | RFLEN2 = 000 0001      2ワード/フレーム   |         |
|      |         |                                    |         |
|      |         |                                    |         |
|      |         | RFLEN1 = 111 1111      128ワード/フレーム |         |
| 7-5  | RWDLEN2 | 受信ワード長2                            | 2.3.4.4 |
|      |         | RWDLEN2 = 000      8ビット            |         |
|      |         | RWDLEN2 = 001      12ビット           |         |
|      |         | RWDLEN2 = 010      16ビット           |         |
|      |         | RWDLEN2 = 011      20ビット           |         |
|      |         | RWDLEN2 = 100      24ビット           |         |
|      |         | RWDLEN2 = 101      32ビット           |         |
|      |         | RWDLEN2 = 11X      予約              |         |

表2-8. 受信制御レジスタ2(RCR2)のビット/フィールドの説明 (続き)

| ビット | 名前       | 機能                                                                   | 節       |
|-----|----------|----------------------------------------------------------------------|---------|
| 4-3 | RCOMPAND | RCOMPAND 受信伸長モード。00b以外のモードは、対応するRWDLENが、8ビット・データを示す000bである場合にのみイネーブル | 2.4     |
|     |          | RCOMPAND = 00 伸長なし。データ転送はMSBから開始されます。                                |         |
|     |          | RCOMPAND = 01 伸長なし、8ビット・データ。転送はLSBから開始されます。                          |         |
|     |          | RCOMPAND = 10 受信データについてμ則により伸長                                       |         |
|     |          | RCOMPAND = 11 受信データについてA則により伸長                                       |         |
| 2   | RFIG     | 受信フレーム無視                                                             | 2.3.6.2 |
|     |          | RFIG = 0 2回目以降の受信フレーム同期パルスでも転送を再開                                    |         |
|     |          | RFIG = 1 2回目以降の受信フレーム同期パルスは無視                                        |         |
| 1-0 | RDATDLY  | 受信データ遅延                                                              | 2.3.4.6 |
|     |          | RDATDLY = 00 0ビットのデータ遅延                                              |         |
|     |          | RDATDLY = 01 1ビットのデータ遅延                                              |         |
|     |          | RDATDLY = 10 2ビットのデータ遅延                                              |         |
|     |          | RDATDLY = 11 予約                                                      |         |

図2-7. 送信制御レジスタ1(XCR1)

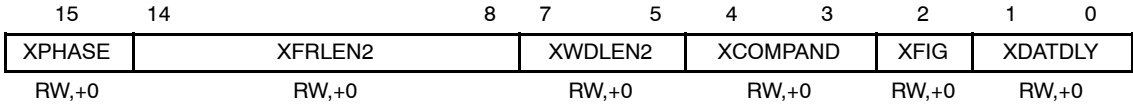


注：R = リード、W = ライト、+0 = リセット時の値

表2-9. 送信制御レジスタ1(XCR1)のビット/フィールドの説明

| ビット  | 名前      | 機能                                  | 節       |
|------|---------|-------------------------------------|---------|
| 15   | 予約      | 予約                                  |         |
| 14-8 | XFRLEN1 | 送信フレーム長1                            | 2.3.4.3 |
|      |         | XFRLEN1 = 000 0000      1ワード/フレーム   |         |
|      |         | XFRLEN1 = 000 0001      2ワード/フレーム   |         |
|      |         |                                     |         |
|      |         |                                     |         |
|      |         | RFRLEN1 = 111 1111      128ワード/フレーム |         |
| 7-5  | XWDLEN1 | 送信ワード長1                             | 2.3.4.4 |
|      |         | XWDLEN1 = 000      8ビット             |         |
|      |         | XWDLEN1 = 001      12ビット            |         |
|      |         | XWDLEN1 = 010      16ビット            |         |
|      |         | XWDLEN1 = 011      20ビット            |         |
|      |         | XWDLEN1 = 100      24ビット            |         |
|      |         | XWDLEN1 = 101      32ビット            |         |
|      |         | XWDLEN1 = 11X      予約               |         |
| 4-0  | 予約      | 予約                                  |         |

図2-8. 送信制御レジスタ2(XCR2)



注： R = リード、W = ライト、+0 = リセット時の値

表2-10. 送信制御レジスタ2(XCR2)のビット/フィールドの説明

| ビット  | 名前      | 機能                                         | 節       |
|------|---------|--------------------------------------------|---------|
| 15   | XP      | 送信フェーズ                                     | 2.3.4.2 |
|      |         | XP = 0                      シングル・フェーズのフレーム |         |
|      |         | XP = 1                      デュアル・フェーズのフレーム |         |
| 14-8 | FRLEN2  | 送信フレーム長2                                   | 2.3.4.3 |
|      |         | FRLEN2 = 000 0000      1ワード/フレーム           |         |
|      |         | FRLEN2 = 000 0001      2ワード/フレーム           |         |
|      |         |                                            |         |
|      |         |                                            |         |
|      |         | FRLEN1 = 111 1111      128ワード/フレーム         |         |
| 7-5  | XWDLEN2 | 送信ワード長2                                    | 2.3.4.4 |
|      |         | XWDLEN2 = 000            8ビット              |         |
|      |         | XWDLEN2 = 001            12ビット             |         |
|      |         | XWDLEN2 = 010            16ビット             |         |
|      |         | XWDLEN2 = 011            20ビット             |         |
|      |         | XWDLEN2 = 100            24ビット             |         |
|      |         | XWDLEN2 = 101            32ビット             |         |
|      |         | XWDLEN2 = 11X            予約                |         |

表2-10. 送信制御レジスタ2(XCR2)のビット/フィールドの説明 (続き)

| ビット | 名前       | 機能                                                         | 節       |
|-----|----------|------------------------------------------------------------|---------|
| 4-3 | XCOMPAND | 送信圧縮モード。00b以外のモードは、適切なXWDLENが、8ビット・データを示す000bである場合にのみイネーブル | 2.4     |
|     |          | XCOMPAND = 00      圧縮なし。データ転送はMSBから開始されます。                 |         |
|     |          | XCOMPAND = 01      圧縮なし、8ビット・データ。転送はLSBから開始されます。           |         |
|     |          | XCOMPAND = 10      送信データについてμ則により圧縮                        |         |
|     |          | XCOMPAND = 11      送信データについてA則により圧縮                        |         |
| 2   | XFIG     | 送信フレーム無視                                                   | 2.3.6.2 |
|     |          | XFIG = 0      2回目以降の送信フレーム同期でも転送を再開                        |         |
|     |          | XFIG = 1      2回目以降の送信フレーム同期は無視                            |         |
| 1-0 | XDATDLY  | 送信データ遅延                                                    | 2.3.4.6 |
|     |          | XDATDLY = 00      0ビットのデータ遅延                               |         |
|     |          | XDATDLY = 01      1ビットのデータ遅延                               |         |
|     |          | XDATDLY = 10      2ビットのデータ遅延                               |         |
|     |          | XDATDLY = 11      予約                                       |         |

## 2.3 データ送受信フロー

2-3ページの図2-1「McBSPの内部ブロック図」に示したように、受信動作はトリプル・バッファ構成、送信動作はダブル・バッファ構成です。受信データはDRピンに入力され、RSR[1,2]にシフト・インされます。ワード中のすべてのビット(8、12、16、20、24、または32ビット)を受信すると、受信バッファ・レジスタRBR[1,2]がフルの状態でない場合に限り、RSR[1,2]がRBR[1,2]にコピーされます。その後、DRR[1,2]がCPUまたはDMAによってリードされていない場合を除き、RBR[1,2]がDRR[1,2]にコピーされます。

送信データは、CPUまたはDMAによってDXR[1,2]にライトされます。XSR[1,2]にデータがない場合は、DXR[1,2]の値がXSR[1,2]にコピーされます。その他の場合は、データの最終ビットがDXピンからシフト・アウトされたときにDXR[1,2]がXSR[1,2]にコピーされます。送信フレーム同期の後、XSR[1,2]がDXからの送信データのシフト・アウトを開始します。

### 2.3.1 シリアル・ポートのリセット: $(R/X)\overline{RST}$ 、および $\overline{RESET}$

シリアル・ポートは、次の2つの方法でリセットできます。

- 1) デバイス・リセット( $\overline{RS} = 0$ )は、受信部、送信部、およびサンプル・レート生成器をリセット状態にします。デバイス・リセットが解除されても( $\overline{RS} = 1$ )、 $\overline{GRST} = \overline{FRST} = \overline{RRST} = \overline{XRST} = 0$ であり、シリアル・ポート全体がリセット状態のままになります。
- 2) シリアル・ポートの送信部と受信部は、シリアル・ポート制御レジスタの $\overline{RRST}$ ビットおよび $\overline{XRST}$ ビットによって個別にリセットできます。サンプル・レート生成器はSPCR2の $\overline{GRST}$ ビットによってリセットされます。

2-23ページの表2-11「McBSPピンのリセット状態」に、デバイス・リセットまたは受信部/送信部リセット( $\overline{XRST} = \overline{RRST} = \overline{FRST} = 0$ )によりシリアル・ポートがリセットされたときのMcBSPピンの状態を示します。



表2-11. McBSPピンのリセット状態

| McBSPピン                                                     | 方向    | デバイス・リセット (RESET = 0) | McBSPリセット                           |
|-------------------------------------------------------------|-------|-----------------------|-------------------------------------|
| 受信部リセット ( $\overline{RRST} = 0$ および $\overline{GRST} = 1$ ) |       |                       |                                     |
| DR                                                          | I     | 入力                    | 入力                                  |
| CLKR                                                        | I/O/Z | 入力                    | 入力の場合は既知の状態、出力の場合はCLKRが動作           |
| FSR                                                         | I/O/Z | 入力                    | 入力の場合は既知の状態、出力の場合はFSRPと同値でインアクティブ状態 |
| CLKS                                                        | I/O/Z | 入力                    | 入力                                  |
| 送信部リセット ( $\overline{XRST} = 0$ および $\overline{GRST} = 1$ ) |       |                       |                                     |
| DX                                                          | O     | ハイ・インピーダンス            | ハイ・インピーダンス                          |
| CLKX                                                        | I/O/Z | 入力                    | 入力の場合は既知の状態、出力の場合はCLKXが動作           |
| FSX                                                         | I/O/Z | 入力                    | 入力の場合は既知の状態、出力の場合はFSXPと同値でインアクティブ状態 |
| CLKS                                                        | I     | 入力                    | 入力                                  |

- デバイス・リセットまたはMcBSPリセット: McBSPが上記2つの方法のどちらかでリセットされると、各ステータスは初期状態にリセットされます。この初期状態では、すべてのカウンタおよびステータス・ビットがリセットされます。受信ステータス・ビットには、RFULL、RRDY、およびRSYNCERRが含まれます。送信ステータス・ビットには、 $\overline{XEMPTY}$ 、XRDY、およびXSYNCERRがあります。
- デバイス・リセット: デバイス・リセットによりMcBSPがリセットされると(デバイス・ピン $\overline{RS} = 0$ )、送信部、受信部、およびサンプル・レート生成器を含むシリアル・ポート全体がリセットされます。すべての入力専用ピンと3ステート・ピンは既知の状態になります。出力専用ピンDXは、ハイ・インピーダンス状態になります。サンプル・レート生成器もリセットされるため( $\overline{GRST} = 0$ )、サンプル・レート生成器クロックCLKGは2分周のCPUクロックによってドライブされるのに対し、フレーム同期信号FSGは生成されません。サンプル・レート生成器リセットの詳細については、2-61ページの第2.5.1.2節「サンプル・レート生成器のリセット手順」を参照してください。デバイスのリセットが解除されても、シリアル・ポートはリセット状態のままとなり( $[R/X]\overline{RST} = 0$ および $\overline{FRST} = 0$ )、この状態では、2-96ページの第2.9節「McBSPの汎用入出力ポート」で説明するように、DRピンとDXピンを汎用I/Oとして使用できます。
- McBSPリセット: 受信部と送信部のリセット・ビット、 $\overline{RRST}$ と $\overline{XRST}$ にゼロがライトされると、McBSPのそれぞれの部分がリセットされ、シリアル・ポートの対応する機能ブロックの動作が停止します。DRやCLKSなどのすべての入力専用ピンと、入力として設定されている他のすべてのピンは、既知の状態になります。FS(R/X)は、出力の場合はインアクティブ状態になります(その極性ビットFS[R/X]Pと同じ)。CLK(R/X)が出力として

プログラムされている場合は、 $\overline{\text{GRST}} = 1$ という条件で、CLK(R/X)がCLKGによってドライブされます。最後に、送信部またはデバイスがリセットされると、DXピンがハイ・インピーダンス状態になります。通常の動作では、サンプル・レート生成器は $\overline{\text{GRST}}$ ビットにゼロをライトすることでリセットできます。 $\overline{\text{GRST}}$ ビットは、送信部または受信部のどちらもサンプル・レート生成器を使用していない場合にのみローにします。この場合、内部サンプル・レート生成器クロック(CLKG)とそのフレーム同期信号(FSG)はインアクティブ状態のローになります。サンプル・レート生成器がリセット状態でない場合( $\overline{\text{GRST}} = 1$ )、FSRピンとFSXピンは、FSGによってドライブされる出力であっても、それぞれ $\overline{\text{RRST}} = 0$ および $\overline{\text{XRST}} = 0$ のときはインアクティブ状態になります。これによって、McBSPの送受信のうちどちらか一方だけがリセットされても、 $\overline{\text{FRST}} = 1$ でフレーム同期がFSGによって駆動されていれば、残りの部分は動作を継続できます。サンプル・レート生成器リセットの詳細については、2-61ページの第2.5.1.2節「サンプル・レート生成器のリセット手順」を参照してください。

- サンプル・レート生成器リセット: 前に述べたように、サンプル・レート生成器はデバイス・リセットまたはそのリセット・ビット $\overline{\text{GRST}}$ にゼロがライトされたときにリセットされます。デバイス・リセットの場合、サンプル・レート生成器クロックCLKGは2分周のCPUクロックによって生成されるのに対し、フレーム同期パルスFSGはインアクティブ状態のローにドライブされます。送信部または受信部のどちらにもCLKGおよびFSGが供給されていないときにサンプル・レート生成器をリセットしたい場合は、SRGR2の $\overline{\text{GRST}}$ をゼロに設定します。この場合、CLKGとFSGはインアクティブ状態のローにドライブされます。 $\overline{\text{GRST}} = 1$ のときは、CLKGはSRGR1でプログラムされたとおりに起動します。その後、 $\overline{\text{FRST}} = 1$ であれば、プログラムされたフレーム周期(FPER + 1)をCLKGサイクルでカウント後、FSGがアクティブ状態のハイになります。

デバイス・リセット完了( $\overline{\text{RS}} = 1$ )後の、シリアル・ポート初期化手順は次のとおりです。

- 1) SPCR[1,2]で $\overline{\text{XRST}} = \overline{\text{RRST}} = \overline{\text{FRST}} = 0$ に設定します。デバイス・リセットを解除した後の場合は、この手順は必要ありません。
- 2) シリアル・ポートがリセット状態( $\overline{\text{XRST}} = \overline{\text{RRST}} = \overline{\text{FRST}} = 0$ )のときに、2-5ページの表2-2「McBSPのレジスタ」にリストされたMcBSP設定レジスタだけを必要に応じてプログラムします(データ・レジスタはプログラムしません)。
- 3) 2ビット・クロック待ちます。これは、内部で正しい同期が行われるようにするためです。
- 4) DXRへのライトなど、必要に応じてデータ送受信のためのセットアップをします。
- 5)  $\overline{\text{XRST}} = \overline{\text{RRST}} = 1$ に設定してシリアル・ポートをイネーブルにします。この時点でSPCR[1,2]にライトされる値は、リセット・ビットが1に変更されただけであり、残りのビット/フィールドは上の手順2と同じ値であることに注意してください。

- 6) 内部生成のフレーム同期が必要な場合は、 $\overline{\text{FRST}} = 1$ に設定します。
- 7) 受信部と送信部がアクティブになるまで、2ビット・クロック待ちます。

あるいは、上記手順1および5のライト時に希望のビットを変更することで、送信部と受信部を個別にリセット状態にするか、リセットを解除することができます。 $\overline{\text{XRST}}$ ビットまたは $\overline{\text{RRST}}$ ビットのアクティブ状態のローの所要サイクルは、少なくとも2ビット・クロック (CLKR/CLKX)を必要とします。

リセット初期化のための上記手順は、受信部または送信部を正常な動作中にリセットする必要がある場合でも、サンプル・レート生成器がどちらの動作にも使用されない場合でも、全般的に適用できます。

注:

(a) シリアル・ポート設定レジスタ、SPCR[1,2]、PCR、RCR[1,2]、XCR[1,2]、およびSRGR[1,2]をユーザーが変更する場合、影響を受けるシリアル・ポートの部分がリセット状態のときに行ってください。

(b) データ送信レジスタDXR[1,2]は、送信部がリセット状態でない( $\overline{\text{XRST}} = 1$ )ときにだけCPUまたはDMAによってロードされるようにします。ただし、デジタル・ループ・バック・モードの場合は例外となります。これについては2-55ページの第2.4.1節「内部データの圧伸」で説明します。

(c) マルチチャネル選択レジスタ、MCR、XCER[A/B]、およびRCER[A/B]は、マルチチャネル選択で現在そのブロックが使用されていない限り、いつでも変更できます。詳細については、2-82ページの第2.6.3.2節を参照してください。

例2-1に、受信部動作中に送信部をリセットおよび設定する場合の制御レジスタの値を示します。

## 例2-1. 受信部動作中の送信部のリセットおよび設定

2

|                                  |                                                                                       |
|----------------------------------|---------------------------------------------------------------------------------------|
| SPCR1 = 0x0001<br>SPCR2 = 0x0030 | 送信部のリセット。送信割り込み(CPUへのXINT)がXSYNCERRによって発生します。RINTはRRDYによって発生し、受信部は動作中です。              |
| PCR = 0x0A00                     | FSXはSRGRのFSGMによって決まり、送信クロックは外部ソースによって生成され、受信クロックは引き続きサンプル・レート生成器によって生成されます。           |
| SRGR1 = 0x0001<br>SRGR2 = 0x2000 | CPUクロックの2分周のサンプル・レート生成器クロック(CLKG)が生成されます。DXR[1,2]からXSR[1,2]へのコピーにより送信フレーム同期信号が生成されます。 |
| XCR1 = 0x0840<br>XCR2 = 0x8421   | デュアル・フェーズ・フレーム。第1フェーズには8つの16ビット・ワード、第2フェーズには4つの12ビット・ワードが含まれ、1ビットのデータ遅延が生じます。         |
| SPCR2 = 0x0031                   | 送信部のリセット解除                                                                            |

### 2.3.2 レディー状態の決定

RRDYとXRDYは、それぞれMcBSP受信部および送信部のレディー状態を示します。シリアル・ポートのリード/ライトは、RRDYビットおよびXRDYビットのポーリング、DMAへのイベント(ノーマル・モードではREVTおよびXEVT、A-bisモードではREVTaおよびXEVTa)、またはイベントが発生するCPU割り込み(RINTおよびXINT)によって同期できます。DRR[1,2]のリードとDXR[1,2]へのライトは、RRDYおよびXRDYに影響を及ぼすことに注意してください。

#### 2.3.2.1 受信レディー状態: REVT、RINT、およびRRDY

RRDY = 1は、RBR[1,2]の内容がDRR[1,2]にコピーされ、CPUまたはDMAがそのデータをリードできることを示します。CPUまたはDMAのどちらかによってデータがリードされると、RRDYビットが0にクリアされます。また、デバイス・リセット時またはシリアル・ポート受信部リセット時( $\overline{RRST} = 0$ )には、RRDYビットが0にクリアされ、データの受信とDRR[1,2]へのロードがまだ行われていないことを示します。RRDYはDMAへのMcBSP受信イベント(REVT)を直接発生させます。CPUへのMcBSP受信割り込み(RINT)も、SPCR1でRINTM = 00bであればRRDYによって発生させることができます。

#### 2.3.2.2 送信レディー状態: XEVT、XINT、およびXRDY

XRDY = 1は、DXR[1,2]の内容がXSR[1,2]にコピーされ、DXR[1,2]に新規データ・ワードのロードが可能であることを示します。送信部がリセット状態から復帰する( $\overline{XRST}$ を0から1にセットする)と、XRDYビットも0から1になり、新しいデータに対してDXR[1,2]がレディーであることを示します。CPUまたはDMAによって新しいデータがロードされると、XRDYビットは0にクリアされます。しかし、このデータがDXR[1,2]からXSR[1,2]にコピーされると、XRDYビットは再度0から1になります。これにより、DXピン上でXSR[1,2]がまだシフト・アウトされていなくても、CPUまたはDMAは再びDXR[1,2]にライトできます。XRDYは、DMAへの送信同期イベント(XEVTまたはXEVTa)を直接発生させます。さらに、CPUへの送

信割り込み(XINT)も、SPCR2でXINTM = 00bであればXRDYによって直接発生させることができます。

### 2.3.3 CPU割り込み: (R/X)INT

受信割り込み(RINT)と送信割り込み(XINT)は、シリアル・ポートの状態の変化をCPUに通知します。これらの割り込みの設定には、4つのオプションがあります。これらのオプションは、SPCR[1,2]の受信/送信割り込みモード・ビット/フィールド(R/X)INTMによって設定します。

- 1) (R/X)INTM=00b。SPCR[1,2]の(R/X)RDYビットが0から1に変化するごとに、割り込みを発生させます。RRDYビットとXRDYビットについては、第2.3.2.1と第2.3.2.2節で説明しています。
- 2) (R/X)INTM=01b。フレーム内の16チャンネルのブロック境界(マルチチャンネル選択モードの場合)ごとに割り込みを発生させます。マルチチャンネル選択モード以外の場合にはこの設定は適用されず、割り込みは発生しません。詳細については、2-83ページの第2.6.3.3節「割り込みの更新」を参照してください。
- 3) (R/X)INTM=10b。フレーム同期パルスを検出すると割り込みを発生させます。この場合、送受信部がリセット状態でも割り込みが発生します。これは、受信したフレーム同期パルスをCPUクロックに同期させ、(R/X)INT経由でCPUに通知するという方法で行われます。これについては、2-68ページの第2.5.3.4節「初期化のためのフレーム検出」で説明します。
- 4) (R/X)INTM=11b。フレーム同期エラーの発生時に割り込みを発生させます。他の割り込みモードが選択されている場合でも、(R/X)SYNCERRをリードすることによりこの状態を検出できます。同期エラーの詳細については、第2.3.7.2および第2.3.7.5節を参照してください。

上記オプションのうち2),3),4)の3つは、CPUへの割り込みとして適用されますが、DMAのイベントとしては、適用されません。

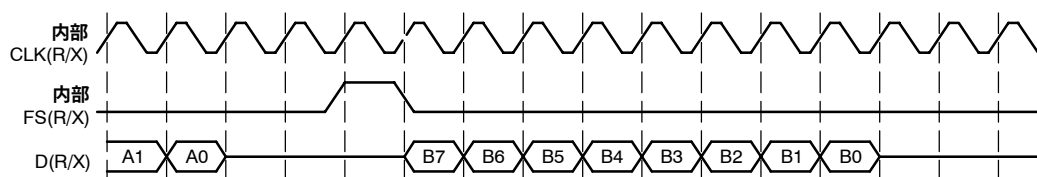
### 2.3.4 フレームおよびクロックの設定

図2-9に、McBSPクロックとフレーム同期信号の一般的な動作を示します。シリアル・クロックのCLKRとCLKXは、それぞれ受信および送信のビット境界を意味します。同様に、フレーム同期信号のFSRとFSXは、シリアル・ワードの先頭を意味します。

McBSPでは、データ・フレーム同期のために様々なパラメータを設定できます。これは受信および送信について個別に設定することができ、次の項目が含まれます。

- ☐ FSR、FSX、CLKX、およびCLKRの極性
- ☐ シングル・フェーズ・フレームまたはデュアル・フェーズ・フレームの選択
- ☐ 各フェーズについて、フレーム当たりのワード数
- ☐ 各フェーズについて、ワード当たりのビット数
- ☐ 後続のフレーム同期について、シリアル・データ・ストリームを再開するか無視するかを選択できます。
- ☐ フレーム同期から最初のデータ・ビットまでのデータ・ビット遅延を、0、1、または2ビットに設定できます。
- ☐ 受信データについては、右詰めまたは左詰めだけでなく、符号拡張またはゼロ詰めも選択できます。

図2-9. フレームおよびクロックの動作



#### 2.3.4.1 フレームおよびクロックの動作

受信フレーム同期パルスと送信フレーム同期パルスは、サンプル・レート生成器(2-58ページの第2.5.1節「サンプル・レート生成器のクロックとフレームの生成」参照)によって内部で生成するか、外部ソースによって駆動させることができます。フレーム同期のソースは、PCRのモード・ビットFS(R/X)Mをプログラムすることで選択できます。FSRはSRGR2のGSYNCビットによっても影響を受けます(詳細については、2-67ページの第2.5.3.2節「受信フレーム同期の選択: DLB、FSRM、GSYNC」を参照のこと)。同様に、PCRのモード・ビットCLK(R/X)Mをプログラムすることで、受信クロックおよび送信クロックを入力または出力として選択できます。

FSRとFSXが入力の場合(FSX=FSRM=0、外部フレーム同期パルス)、McBSPは内部クロック、つまり内部CLKRおよび内部CLKXの立ち下がりエッジでそれぞれを検出します(2-57ページの図2-41「クロックおよびフレームの生成」参照)。DRピンに到達した受信データも、内部CLKRの立ち下がりエッジでサンプリングされます。これらの内部クロック信号はCLK(R/X)ピン経由で外部ソースから供給されるか、McBSP内部のサンプル・レート生成器クロック(CLKG)によって生成されることに注意してください。

FSRとFSXが出力の場合、FSRとFSXはサンプル・レート生成器によって生成され、FSRとFSXは内部クロックCLK(R/X)の立ち上がりエッジで生成されます(アクティブ状態への遷移)。同様に、DXピン上のデータは内部CLKXの立ち上がりエッジで出力されます。詳細については、2-34ページの第2.3.4.6節を参照してください。

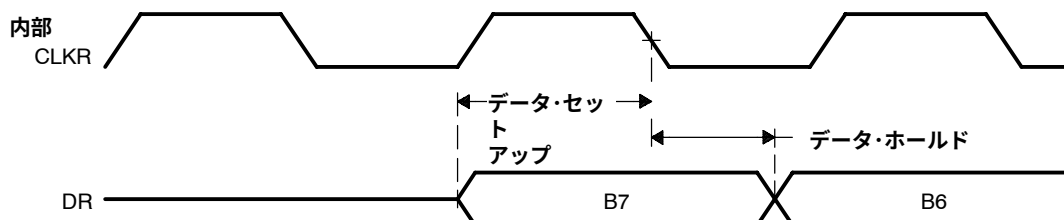
FSRP、FSXP、CLKRP、およびCLKXPビットは、2-12ページの表2-6「ピン制御レジスタ(PCR)のビット/フィールドの説明」に示したように、FSR、FSX、CLKR、およびCLKXの極性を設定します。シリアル・ポート内部のフレーム同期信号(内部FSR、内部FSX)は、すべてアクティブ・ハイです。シリアル・ポートが外部フレーム同期に設定されていて(FSR/FSXがMcBSPへの入力)、FSRP = FSXP = 1であれば、外部のアクティブ・ロー・フレーム同期信号が、受信部(内部FSR)および送信部(内部FSX)に送信される前に反転します。同様に、内部同期(FSR/FSXが出力ピンで、GSYNC = 0)が選択された場合は、極性ビットFS(R/X)P = 1であれば、内部のアクティブ・ハイ同期信号が、FS(R/X)ピンに送信される前に反転します。2-57ページの図2-41に、XORゲートによるこの反転を示します。

送信側では、送信クロックの極性ビットCLKXPにより、送信データのシフト・アウトおよびクロック・アウトに使用されるエッジが設定されます。データは常に内部CLKXの立ち上がりエッジで送信されます。CLKXP = 1で、外部クロックが選択されている場合(CLKXM = 0かつCLKXが入力)は、CLKXピンに入力される外部クロック(立ち下がりエッジ・トリガ)は、送信部に送られる前に反転されます(立ち上がりエッジ・トリガ)。CLKXP = 1で、内部クロックが選択されている場合(CLKXM = 1かつCLKXが出力ピン)は、内部(立ち上がりエッジ・トリガ)クロックCLKXが、CLKXピンから出力される前に立ち下がりエッジに反転されます。

同様に受信部は、(送信器からの)クロックの立ち上がりエッジによって順に送られてくるデータをサンプリングできます。受信クロック極性ビットCLKRPは、受信データのサンプリングに使用されるエッジを設定します。受信データは常に内部CLKRの立ち下がりエッジでサンプリングされます。したがって、CLKRP = 1で外部クロックが選択されている場合(CLKRM = 0かつCLKRが入力ピン)は、CLKRピンに入力される外部クロック(立ち上がりエッジ・トリガ)は受信部に送られる前に反転されます(立ち下がりエッジ・トリガ)。CLKRP = 1で、内部クロックが選択されている(CLKRM = 1)場合は、内部(立ち下がりエッジ・トリガ)クロックが、CLKRピンから出力される前に立ち上がりエッジに反転します。

システム内で受信部と送信部のクロッキングに同じクロック(内部または外部)が使用される場合はCLKRP=CLKXPです。受信部は送信部と逆のエッジを使用して、このエッジに対するデータのセットアップ時間とホールド時間を確保します。図2-10に、外部シリアル・デバイスによって立ち上がりエッジで出力されたデータを、同じクロックの立ち下がりエッジでMcBSP受信部がサンプリングする様子を示します。

図2-10. 受信データとクロック



### 2.3.4.2 フレーム同期フェーズ

フレーム同期は、McBSPによる転送の開始を示します。フレーム同期に続くデータ・ストリームには、2つのフェーズ、つまりフェーズ1とフェーズ2を転送できます。フェーズの数は、フェーズ・ビット、つまりRCR2とXCR2の(R/X)PHASEビットによって選択できます。1フレーム当たりのワード数と1ワード当たりのビット数は、それぞれ(R/X)FRLLEN[1,2]フィールドと(R/X)WDLEN[1,2]フィールドによって、フェーズごとに個別に選択できます。図2-11に、最初のフェーズが2つの12ビット・ワード、続く2番目のフェーズが3つの8ビット・ワードで構成されている例を示します。フレーム内のビット・ストリーム全体が連続していることに注意してください。ワード間にもフェーズ間にもギャップがありません。表2-12に、受信部と送信部の両方について、各フェーズの1フレーム当たりのワード数と1ワード当たりのビット数を制御する、受信/送信制御レジスタ(RCR[1,2]/XCR[1,2])のビット/フィールドを示します。1フレーム当たりの最大ワード数は、シングル・フェーズ・フレームでは128、デュアル・フェーズ・フレームでは256です。1ワード当たりのビット数は、8、12、16、20、24、または32ビットのいずれかとなります。

図2-11. デュアル・フェーズ・フレームの例

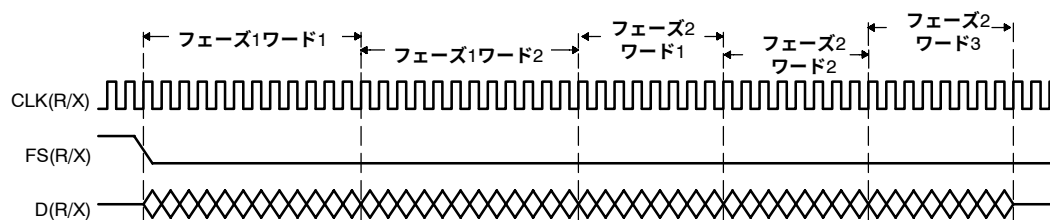




表2-12. 1フレーム当たりのワード数と1ワード当たりのビット数を制御するRCR[1,2]/XCR[1,2]のビット/フィールド

| シリアル・ポート<br>McBSP0/1 | フレーム<br>フェーズ | RCR[1,2]/XCR[1,2]のビット/フィールド制御 |         |
|----------------------|--------------|-------------------------------|---------|
|                      |              | ワード/フレーム                      | ビット/ワード |
| 受信                   | 1            | RFRLEN1                       | RWDLEN1 |
| 受信                   | 2            | RFRLEN2                       | RWDLEN2 |
| 送信                   | 1            | XFRLEN1                       | XWDLEN1 |
| 送信                   | 2            | XFRLEN2                       | XWDLEN2 |

## 2.3.4.3 フレーム長: (R/X)FRLEN[1,2]

フレーム長は、1フレームにつき転送されるシリアル・ワード(8、12、16、20、24、または32ビット)の数として定義できます。この長さは、フレーム同期信号当たりのワードまたは論理タイム・スロットまたはチャンネル数に対応します。表2-13に示すように、(R/X)CR[1,2]の7ビットの(R/X)FRLEN[1,2]フィールドは、1フレーム当たり最大128ワードをサポートします。データ・ストリームとして、(R/X)PHASE = 0ではシングル・フェーズが、(R/X)PHASE = 1ではデュアル・フェーズが選択されます。シングル・フェーズ・フレームについては、FRLEN2は参照されません。フレーム長のフィールドには(w-1)を設定してください。ここでwは、1フレーム当たりのワード数です。図2-11の例では、(R/X)FRLEN1 = 1または0000001bであり、(R/X)FRLEN2 = 2または0000010bとなります。

表2-13. McBSP受信/送信フレーム長(1、2)の設定

| (R/X)<br>PHASE | (R/X)FRLEN1         | (R/X)FRLEN2         | フレーム長                                   |
|----------------|---------------------|---------------------|-----------------------------------------|
| 0              | $0 \leq n \leq 127$ | X                   | シングル・フェーズ・フレーム: フレーム当たり (n+1) ワード       |
| 1              | $0 \leq n \leq 127$ | $0 \leq m \leq 127$ | デュアル・フェーズ・フレーム: フレーム当たり (n+1)+(m+1) ワード |

## 2.3.4.4 ワード長: (R/X)WDLEN[1,2]

受信/送信制御レジスタの3ビットの(R/X)WDLEN[1,2]フィールドは、表2-12に示したように、フレームの各フェーズについて、受信部および送信部のワード長を、ビット/ワードを単位として決定します。表2-14に、各ワード長に対応するこれらのフィールドの設定値を示します。

2-30ページの図2-11の例では、(R/X)WDLEN1 = 001b、(R/X)WDLEN2 = 000bです。

注：

(a) シングル・フェーズ・フレームを示す(R/X)PHASE = 0に設定すると、(R/X)WDLEN2は使用されず、その値は参照されません。

(b) 指定したワード長が16ビットを超える場合は、D(X/R)R2をD(X/R)R1よりも前にリード/ライトする必要があります。

表2-14. McBSP受信/送信ワード長の設定

| (R/X)WDLEN[1,2] | McBSPワード長(ビット) |
|-----------------|----------------|
| 000             | 8              |
| 001             | 12             |
| 010             | 16             |
| 011             | 20             |
| 100             | 24             |
| 101             | 32             |
| 110             | 予約             |
| 111             | 予約             |

#### 2.3.4.5 フレーム長およびワード長による効率的なデータ転送

フレーム長とワード長を操作して、効率的なデータ転送ができます。たとえば、図2-12のようにシングル・フェーズ・フレームで4つの8ビット・ワードが転送される状況を考えてます。この場合、次のように設定します。

- ☐ (R/X)FRLLEN1 = 0000011b、4ワード・フレーム
- ☐ (R/X)PHASE = 0、シングル・フェーズ・フレーム
- ☐ (R/X)FRLLEN2 = X
- ☐ (R/X)WDLEN1 = 000b、8ビット・ワード

この場合、4つの8ビット・データ要素がCPUまたはDMAによってMcBSPに、またはMcBSPから転送されます。したがって、フレームごとにDRR1からのリードが4回、DXR1へのライトが4回必要になります。

図2-12. 4つの8ビット・ワードで構成されるシングル・フェーズ・フレーム

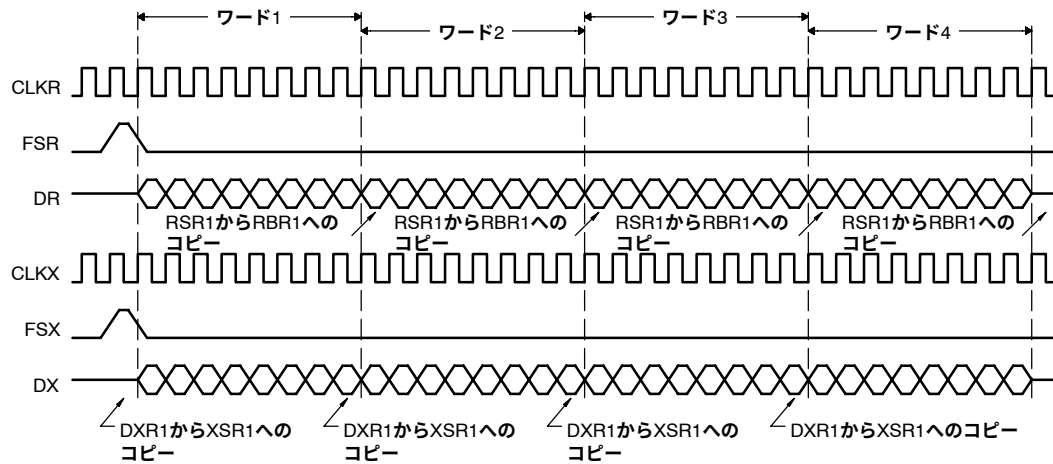


図2-12の例は、図2-13に示すように、1つの32ビット・データ・ワードで構成されるシングル・フェーズ・フレームのデータ・ストリームとして扱うこともできます。この場合、次のように設定します。

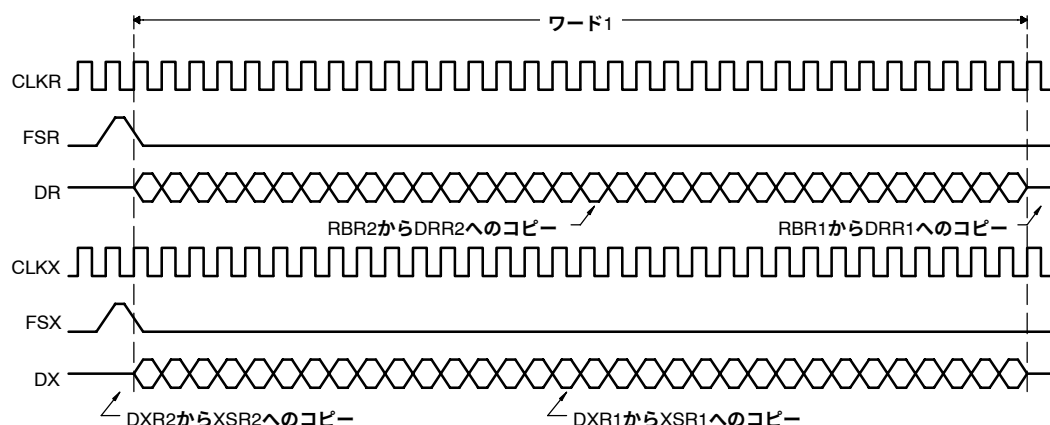
- ☐ (R/X)FRLLEN1 = 0b、1ワード・フレーム
- ☐ (R/X)PHASE = 0、シングル・フェーズ・フレーム
- ☐ (R/X)FRLLEN2 = X
- ☐ (R/X)WDLEN1 = 101b、32ビット・ワード

この場合、CPUまたはDMAによって2つの16ビット・データ・ワードがMcBSPに、またはMcBSPから転送されます。したがって、フレームごとにDRR2およびDRR1からのリードが2回、DXR2およびDXR1へのライトが2回必要になります。その結果、前の場合に比べ転送回数が1/2になります。この操作により、シリアル・ポートによるデータ転送において、バスの専有時間を減らすことができます。

注：

この場合、D(X/R)R2をD(X/R)R1よりも前にリード/ライトする必要があります。

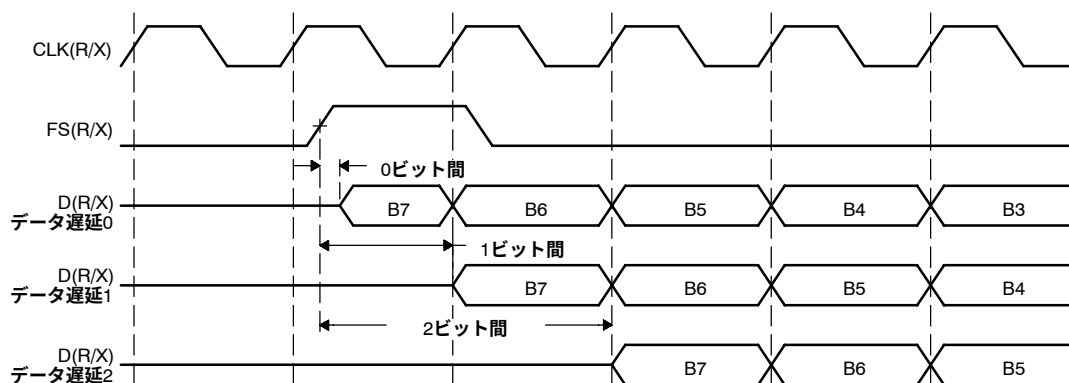
図2-13. 1つの32ビット・ワードで構成されるシングル・フェーズ・フレーム



## 2.3.4.6 データ遅延: (R/X)DATDLY

フレームの開始は、フレーム同期がアクティブであることが認識された最初のクロック・サイクルと定義されます。必要な場合は、実際のデータ受信または送信の開始を、フレームの開始に対して遅延させることができます。この遅延をデータ遅延といいます。RDATDLYとXDATDLYは、それぞれ受信および送信のデータ遅延を設定します。プログラム可能なデータ遅延の範囲は、2-16ページの表2-7「受信制御レジスタ1(RCR1)のビット/フィールドの説明」、2-17ページの表2-8「受信制御レジスタ2(RCR2)のビット/フィールドの説明」、および図2-14「データ遅延」に示したように、0~2ビット・クロック ( $(R/X)DATDLY = 00b - 10b$ )です。一般には1ビットの遅延を選択します。これは、データは1サイクルのアクティブなフレーム同期パルスに続くことが多いからです。

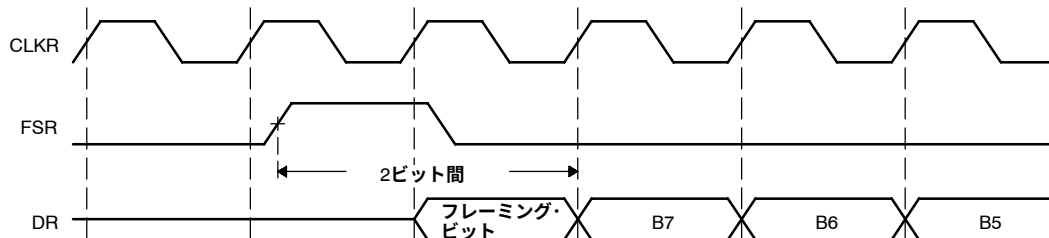
図2-14. データ遅延



通常、フレーム同期パルスは内部シリアル・クロックCLK(R/X)のエッジに対して検出またはサンプリングされます(2-28ページの第2.3.4.1節「フレームおよびクロックの動作」参照)。したがって、次のサイクル以降(データ遅延の値による)でデータの受信または送信ができます。しかし、ゼロ・ビットのデータ遅延の場合は、同じシリアル・クロック・サイクルの中でデータを受信または送信する準備ができていなければなりません。受信については、この問題は解決されています。というのは、受信データは内部FSRがアクティブ(ハイ)の時に、CLKRの最初の立ち下がりエッジで検出されるためです。しかし、データ送信はフレーム同期を生成した内部CLKXクロックの立ち上がりエッジで開始する必要があります。したがって、最初のデータ・ビットはXSR1にあることを前提とします。送信部はその後、フレーム同期FSXがアクティブ(ハイ)になるのを非同期的に検出し、DXピンに最初のビットの送信を即座に開始します。

一般的に使用されるもう1つのモードは、2ビットのデータ遅延です。この設定では、データ・ストリームの前にフレーミング・ビットを送信する各種のT1フレーミング・デバイスをシリアル・ポートに直接インターフェイスできます。データ遅延を2ビットに設定してこのようなストリームを受信する(フレーミング・ビットは1ビットの遅延の後に現れ、データは2ビットの遅延の後に現れます)と、実際にはシリアル・ポートは図2-15のように、データストリームのフレーミング・ビットを破棄することになります。送信では、シリアル・ポートは最初の転送ビットを遅延させることで、実際にはフレーミング・ビットの代わりに空白の期間(ハイ・インピダンス期間)を挿入します。この場合、フレーミング・デバイスが独自のフレーミング・ビットを挿入するか、別のデバイスによってフレーミング・ビットを生成させることになります。あるいは、DXをプルアップまたはプルダウンして希望の値にすることもできます。

図2-15. フレーミング・ビット破棄に使用される2ビットのデータ遅延



#### 2.3.4.7 複数フェーズ・フレームの例: AC97

図2-16に、デュアル・フェーズ・フレーム機能を使用するAudio Codec '97(AC97)規格の例を示します。最初のフェーズは、1つの16ビット・ワードで構成されています。2番目のフェーズは、12の20ビット・ワードで構成されています。これらのフェーズは次のように設定します。

- ☐ (R/X)PHASE = 1b、デュアル・フェーズ・フレーム
- ☐ (R/X)FRLN1 = 0b、フェーズ1ではフレーム当たり1ワード

- ☐ (R/X)WDLEN1 = 010b、フェーズ1ではワード当たり16ビット
- ☐ (R/X)FRLLEN2 = 0001011b、フェーズ2ではフレーム当たり12ワード
- ☐ (R/X)WDLEN2 = 011b、フェーズ2ではワード当たり20ビット
- ☐ CLK(R/X)P = 0、受信データは内部CLKRの立ち下がりエッジで検出されます。送信データは内部CLKXの立ち上がりエッジで送信されます。
- ☐ FS(R/X)P = 0、アクティブ・ハイのフレーム同期信号
- ☐ (R/X)DATDLY = 01b、1ビット・クロックのデータ遅延

図2-16. AC97のデュアル・フェーズ・フレーム・フォーマット

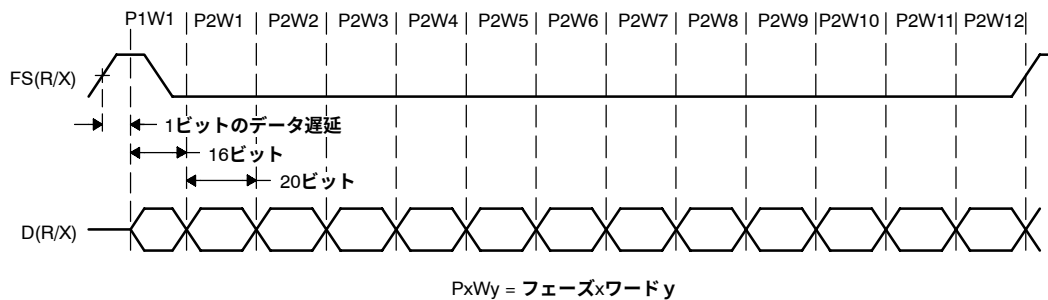
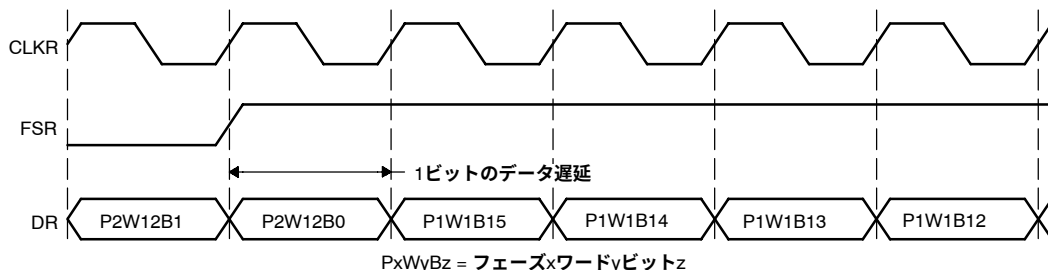


図2-16は、フレーム同期付近のAC97のタイミングを示しています。まず、フレーム同期パルス自体が最初のワードにオーバーラップしていることに注意してください。McBSPの動作では、フレーム同期信号のインアクティブからアクティブへの遷移が、実際のフレーム同期を意味します。このため、フレーム同期は任意のビット数だけハイにすることができます。フレーム同期がインアクティブになり再度アクティブになったことが認識されて初めて、次のフレーム同期が認識されます。

また、図2-17では1ビットのデータ遅延があることに注意してください。データ遅延があるにもかかわらず、送信はギャップなしで行うことができます。フェーズ2における前(最後)のワードの最終ビットが終わると、次のデータ・フレームのフェーズ1における最初のワードの最初のデータ・ビットがすぐに続きます。

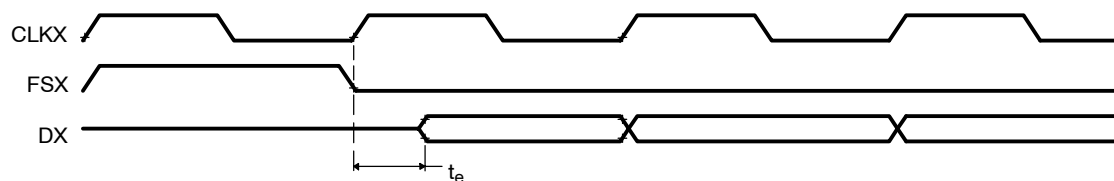
図2-17. フレーム同期付近のAC97ビット・タイミングの例



### 2.3.4.8 DXピンにおける遅延イネーブル/ディセーブル

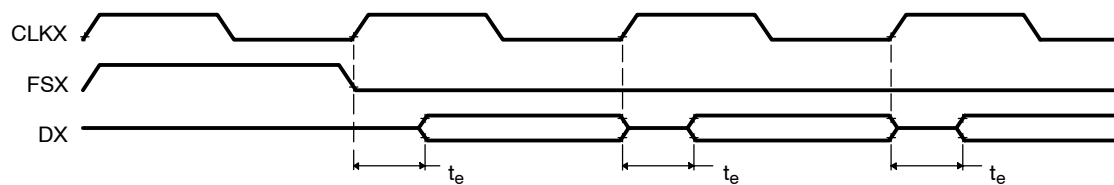
図2-18と図2-19に、DXENAビットを1に設定して、送信データのターンオン時間への追加遅延をイネーブルにした場合のDXピンのタイミングを示します。このビットは、データ自体ではなくDXピン上のハイ・インピーダンス(hi-Z)イネーブルを制御します。したがって、ノーマル・モードでは最初のビットだけが遅延されます。A-bisモードでは、あらゆるビットがハイ・インピーダンスからデータが有効な状態に遷移できるため、あらゆるビットを遅延させることができます。このビットは、DXピンを結線する場合のデータ衝突を回避するために使われます。

図2-18. ノーマル・モードにおけるDXイネーブラ



注:  $t_e$  = (DXENA = 1)による、ターンオン時間に対する追加遅延

図2-19. A-bisモードでのDXイネーブラ



注:  $t_e$  = (DXENA = 1)による、ターンオン時間に対する追加遅延

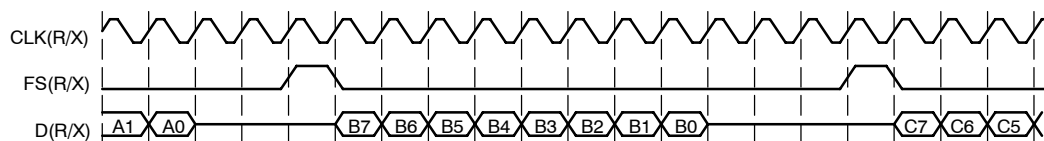
### 2.3.5 McBSPの基本的な動作

シリアル転送中は、多くの場合パケットまたは転送の間にシリアル・ポートのインアクティブ期間があります。受信フレーム同期パルスと送信フレーム同期パルスは、シリアル転送が行われるたびに生成されます。シングル・フェーズ・フレームの(R/X)PHASE = 0とし、必要なワード数を(R/X)FRLLEN1でプログラムし、McBSPが目的の動作に設定されている場合、リセットが解除されるとシリアル転送を開始できます。ワード数の範囲は、1から128まで ( $(R/X)FRLLEN1 = 0x0 \sim 0x7F$ ) です。シリアル・ワード長は、(R/X)CR1の(R/X)WDLEN1 フィールドで設定できます。転送にデュアル・フェーズが必要な場合、つまり (R/X)PHASE = 1 の場合は、(R/X)WDLEN1[1,2]を0x0から0x7Fまでの任意の値に設定できます。これは、1から128までを表します。

図2-20に、1つの8ビット・ワードで構成されるシングル・フェーズ・データ・フレームの例を示します。転送には1データ・ビットの遅延が設定されているため、DXピンおよびDRピン上のデータは、FS(R/X)がアクティブになってから1ビット・クロック後に送受信を開始します。この図も含めて、この節では次のような設定条件での動作を例とします。

- ☐ (R/X)FRLLEN1 = 0b、1フレーム当たり1ワード
- ☐ (R/X)PHASE = 0、シングル・フェーズ・フレーム
- ☐ (R/X)FRLLEN2 = X、(R/X)WDLEN2 = X、設定値は参照されません
- ☐ (R/X)WDLEN1 = 000b、8ビット・ワード
- ☐ CLK(R/X)P = 0、受信データは立ち下がりエッジで検出されます。送信データは立ち上がりエッジで送信されます。
- ☐ FS(R/X)P = 0、アクティブ・ハイのフレーム同期信号
- ☐ (R/X)DATDLY = 01b、1ビットのデータ遅延

図2-20. McBSPの基本的な動作



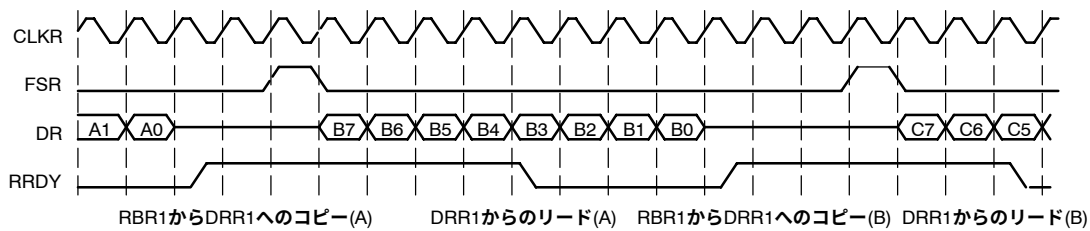
#### 2.3.5.1 受信動作

図2-21に、シリアル受信の例を示します。受信フレーム同期(FSR)がアクティブ状態になると、受信部のCLKRの最初の立ち下がりエッジで検出されます。DRピン上のデータは、RDATDLYによって設定されたデータ遅延の後、受信シフト・レジスタ(RSR[1,2])にシフトされます。RSR[1,2]の内容は、RBR[1,2]が前のデータでフル状態でない限り、クロックの立ち上がりエッジでワード終端ごとにRBR[1,2]にコピーされます。そして、次のCLKRの立ち下がりエッジでのRBR[1,2]からDRR[1,2]へのコピーにより、RRDYステータス・ビットが1(アク



ティブ) にセットされます。これは、受信データ・レジスタ(DRR[1,2])がCPUまたはDMAによるデータ・リードの準備ができていることを示します。DRR[1,2]がCPUまたはDMAによってリードされると、RRDYはインアクティブになります。

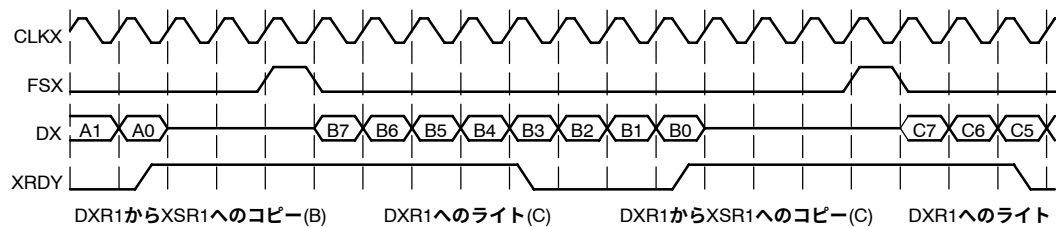
図2-21. 受信動作



### 2.3.5.2 送信動作

送信フレーム同期が生成されると、XDATDLYビットで設定されたデータ遅延の後、送信シフト・レジスタXSR[1,2]の値がシフト・アウトされ、DXピン上に出力されます。XRDYビットは、DXR[1,2]からXSR[1,2]へのコピーが行われるたびに、CLKXの次の立ち上がりエッジでアクティブになります。これは、次に送信するデータがデータ送信レジスタ(DXR[1,2])にライトできることを示します。CPUまたはDMAによってDXR[1,2]へライトされると、XRDYビットはインアクティブになります。図2-22に、シリアル送信の例を示します。送信部のリセットが解除されたとき( $\overline{\text{XRSTY}} = 1$ )の送信動作については、2-48ページの第2.3.7.4節「送信エンプティ: XEMPTY」を参照してください。

図2-22. 送信動作



### 2.3.5.3 最大フレーム周波数

フレーム周波数は、フレーム同期信号の間隔によって決まります。

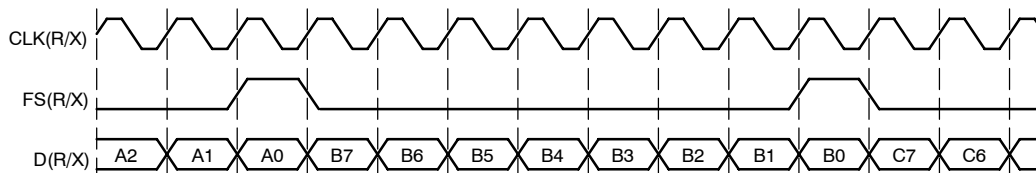
$$\text{フレーム周波数} = \frac{\text{ビット・クロック周波数}}{\text{フレーム同期信号間のビット・クロック数}}$$

フレーム周波数は、フレーム同期信号の間のビット・クロック数(フレーム当たりのビット数によって制限されます)を少なくすることで、高くなります。フレームの送信周波数が高くなると、隣接する転送データ・パケット間のインアクティブ期間がゼロに近づいていきます。フレーム同期の最小間隔は、1フレーム当たりの転送ビット数となります。最大フレーム周波数は次のように定義されます。

$$\text{最大フレーム周波数} = \frac{\text{ビット・クロック周波数}}{\text{1フレーム当たりのビット数}}$$

図2-23に、最大パケット周波数で動作している時のMcBSPを示します。最大パケット周波数では、連続するパケットのデータ・ビットは、ビット間のインアクティブ期間なしで連続的に送信されます。図のように1ビットのデータ遅延がある場合は、フレーム同期パルスが、前のフレームで送信された最終ビットに重なります。

図2-23. 最大フレーム周波数での受信/送信((R/X)DATDLY = 1)



これによって、連続したデータ・ストリームが可能になりますが、この場合のフレーム同期パルスは実際には冗長になります。理論上、マルチパケット転送を開始するために必要なのは、最初のフレーム同期パルスだけです。McBSPは、後続のフレーム同期パルスを無視することで、このようなシリアル・ポートの動作をサポートします。データはクロックごとに、同期して受信部に入力されるか、送信部から出力されます<sup>†</sup>。(R/X)CRのフレーム無視ビット(R/X)FIGをプログラムして、目的のフレーム長またはワード数の送受信が完了するまで後続のフレーム同期パルスを無視することができます。これについては、2-41ページの第2.3.6.1節「フレーム同期無視ビットによる効率的なデータ転送」で説明します。

### 2.3.6 フレーム同期の無視

McBSPは、送信フレーム同期パルスと受信フレーム同期パルスを無視するよう設定できます。(R/X)CR2の(R/X)FIGビットを0に設定してフレーム同期パルスを認識させるか、1に設定してフレーム同期パルスを無視させることができます。ユーザーは(R/X)FIGビットを使って、データを一まとめにする、あるいは予定外のフレーム同期パルスを無視することができます。第2.3.6.1節では、データを一まとめにすることで効率的なデータ転送を行う方法と、予定外のフレーム同期パルスに対するMcBSPの動作について説明します。

<sup>†</sup> (R/X)DATDLY = 0の場合、送信されるデータの先頭ビットは内部CLKXに同期されません。

### 2.3.6.1 フレーム同期無視ビットによる効率的なデータ転送

2-32ページの第2.3.4.5節「フレーム長およびワード長による効率的なデータ転送」では、ワード長およびフレーム長を変更して32ビット・シリアル・ワード転送に設定することで、転送におけるバスの専有時間を減らす1つの方法を説明しました。この例は、フレーム当たりのワード数が複数である場合に利用できます。ここでは、図2-24のようにMcBSPが最大パケット周波数で動作する場合を考えます。この場合、各フレームには8ビット・ワードが1つしかありません。このストリームでは、8ビット・ワードごとに1回のリードと1回のライトが必要です。図2-25に、このストリームを連続した32ビットのワードとして扱うよう設定されたMcBSPを示します。この例では(R/X)FIGが、最初のフレーム以降のフレームを無視するよう1に設定されています。32ビットごとに、リードとライトがそれぞれ2回だけで実行できます。この設定では、4つの8ビット・ワードを転送する場合に比べ、バスの専有時間が事実上1/2に縮小されます。

2

図2-24. 8ビット・データでの最大パケット周波数動作

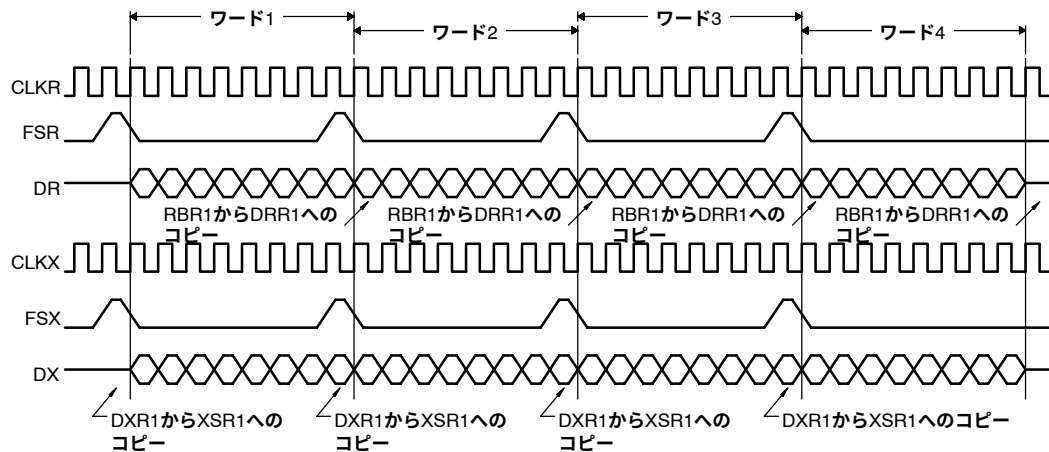
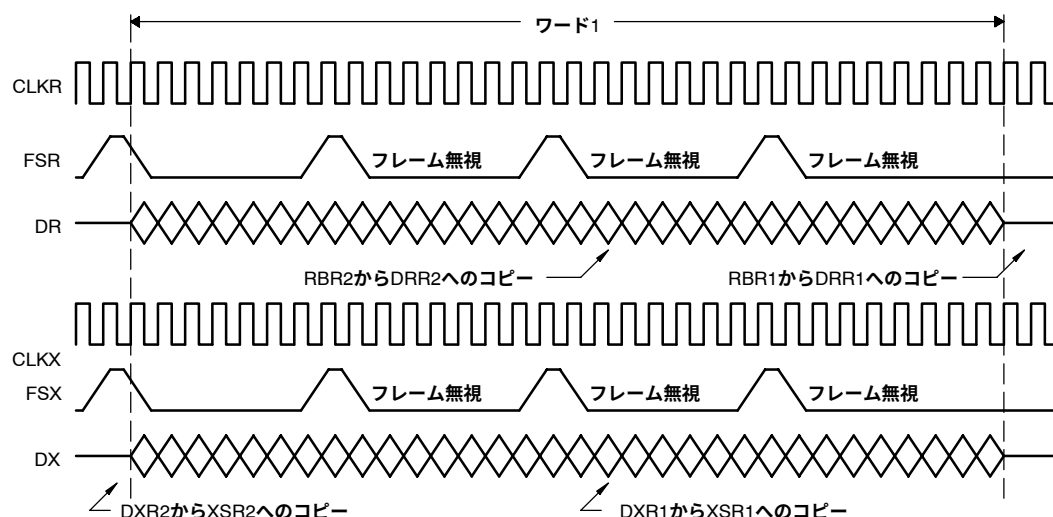


図2-25. (R/X)FIG=1の場合の最大パケット周波数での効率的なデータ転送



### 2.3.6.2 フレーム同期の無視と予定外のフレーム同期パルス

前節では、フレーム無視ビットを使ってデータをまとめ、バスを効率よく使用する方法を説明しました。(R/X)FIGビットは、予定外のフレーム同期パルスが無視するためにも使用できます。ここで、前のフレームの最終受(送)信ビットの前にプログラムされたデータ遅延([R/X]DATDLY)よりも早く発生した同期パルスは、予定外とみなされます。フレーム無視ビットを1に設定すると、シリアル・ポートはそのような予定外のフレーム同期信号を無視します。

受信では、予定外のFSRパルスが無視されない場合(RFIG = 0)、その予定外のパルスにより、それ以降の新しい着信データを優先してRSR[1,2]の内容は破棄されます。したがってRFIG = 0であれば、予定外のフレーム同期パルスによって現在のデータ転送が中止され、SPCR1のRSYNCERRビットが1にセットされ、新しいデータ・ワードの転送が開始されます。詳細については、2-46ページの第2.3.7.2節「**予定外の受信フレーム同期: RSYNCERR**」を参照してください。RFIG = 1の場合は受信が継続され、予定外のフレーム同期パルスは無視されます。

送信では、予定外のFSXパルスが無視されない場合(XFIG = 0)、そのパルスによって現在の送信が中止され、SPCR2のXSYNCERRビットが1にセットされ、中止された現在のワードが再送信されます。詳細については、2-50ページの第2.3.7.5節「**予定外の送信フレーム同期: XSYNCERR**」を参照してください。XFIG = 1の場合は送信が継続され、予定外のフレーム同期信号は無視されます。

図2-26. (R/X)FIG = 0での予定外のフレーム同期

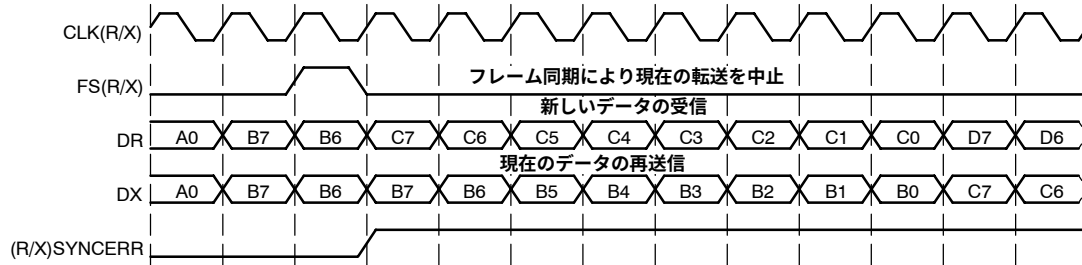
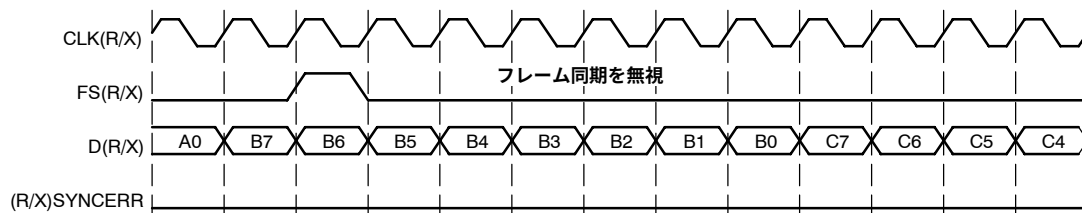


図2-26に、(R/X)FIG = 0の場合に予定外のフレーム同期パルスによりワードBの転送が妨げられる例を示します。受信の場合は、Bの受信が中止され(Bは失われます)、設定されたデータ遅延の後に新しいデータ・ワード(この例ではC)が受信されます。この状態は受信同期エラーであるため、RSYNCERRビットが1にセットされます。しかし送信の場合は、Bの送信が中止され、設定されたデータ遅延の後に同じデータ(B)が再送信されます。この状態は送信同期エラーであるため、XSYNCERRビットが1にセットされます。同期エラーについては、第2.3.7.2節および第2.3.7.5節で説明します。これに対し、図2-27は(R/X)FIG = 1に設定して予定外のフレーム同期信号を無視した場合のMcBSPの動作を示します。ここでは、ワードBの転送が予定外のフレーム同期による影響を受けません。

図2-27. (R/X)FIG = 1での予定外のフレーム同期



### 2.3.7 シリアル・ポートの例外条件

システム・エラーを引き起こす可能性のあるシリアル・ポート・イベントが5つあります。

- 1) 受信オーバーラン(RFULL = 1)。これは、RBR[1,2]からDRR[1,2]への最後のコピーの後、DRR[1,2]がリードされていない場合に起こります。その結果、RBR[1,2]の新しいワードがDRR[1,2]に転送されず、RSR[1,2]はDRピンからシフト・インされた別の新しいワードでフル状態になります。したがってRFULLは、この時点で新しいデータがDRに到着するとRSR[1,2]の内容が上書きされ、前のワードが失われることを示します。新しいデータがDRに到着し、DRR[1,2]がリードされない限り、RSR[1,2]は上書きされ続けます。

- 2) 予定外の受信フレーム同期(RSYNCERR = 1)。これは、RFIG = 0で予定外のフレーム同期パルスが発生したことを示します。予定外のフレーム同期パルスは、前のフレームの最終受信ビットの前に、RDATDLYビット・クロックよりも早く発生したフレーム同期パルスを指します。これによって現在のデータ受信が中止され、リスタートされます。すでにRBR[1,2]からDRR[1,2]へのコピーが行われていて、新しいデータ(予定外のフレーム同期パルスにより中断されたデータ)がRSR[1,2]からRBR[1,2]にコピーされても、このRBR[1,2]中のデータは失われます。これは受信が予定外のフレーム同期パルスにより、リスタートされた場合はそのデータのRBR[1,2]からDRR[1,2]へのコピーが行われなからです。
- 3) 送信データ上書き。これは、DXR[1,2]のデータがXSR[1,2]にコピーされる前にユーザーがそのデータを上書きするものです。前にDXR[1,2]にあったデータはXSR[1,2]にコピーされないため、DXに転送されません。
- 4) 送信エンプティ( $\overline{\text{XEMPTY}} = 0$ )。新しいデータがDXR[1,2]にロードされる前に新しいフレーム同期信号が到着すると、DXR[1,2]の古いデータが再送信されます。これは、DXR[1,2]に新しいデータがロードされるまで、新しいフレーム同期信号がFSXピンに到着するたびに繰り返されます。
- 5) 予定外の送信フレーム同期(XSYNCERR = 1)。これは、XFIG = 0で予定外の同期パルスが発生した場合に起こることがあります。この場合も、予定外のフレーム同期パルスは、前のフレームの最終送信ビットの前にXDATDLYビットクロックよりも早く発生したフレーム同期パルスを示します。これによって現在のデータ送信が中止され、現在の送信が再び開始されます。すでにDXR[1,2]からXSR[1,2]へのコピーが行われていて、DXR[1,2]に新しいデータをライトした場合でも、XSR[1,2]中の送信途中の残りのデータは失われます。

これらのイベントについては、以下の各節で詳細に説明します。

#### 2.3.7.1 オーバーランをとまなう受信: RFULL

SPCR1のRFULL = 1は、受信部でオーバーランが発生し、エラー状態であることを示します。RFULLビットは、次の条件がすべて満たされたときにセットされます。

- 1) RBR[1,2]からDRR[1,2]への最後の転送以降にDRR[1,2]がリードされていない(RRDY = 1)。
- 2) RBR[1,2]が一杯であり、RBR[1,2]からDRR[1,2]へのコピーが行われていない。
- 3) RSR[1,2]が一杯であり、RSR[1,2]からRBR[1,2]への転送が行われていない。

DRに到着するデータは、連続してRSR[1,2]にシフト・インされます。ワード中の全てのビットがRSR[1,2]にシフト・インされると、RBR[1,2]からDRR[1,2]へのコピーが完了している場合にのみRSR[1,2]からRBR[1,2]への転送が行われます。したがって、RBR[1,2]からDRR[1,2]への最後の転送以降にCPUまたはDMAによってDRR[1,2]がリードされていない場合(RRDY = 1)は、RBR[1,2]からDRR[1,2]へのコピーはRRDY = 0になるまで行われません。この時点で、DRピンに到着した新しいデータがRSR[1,2]にシフト・インされ、RSR[1,2]の前の内容は失われます。このデータ損失が起こるのは、シリアル・ワードの受信が完了したとき、RRDY = 0の場合にしかRBR[1,2]からDRR[1,2]への転送が行われないからです。受信部分がリセット状態から起動した後、少なくとも3ワードを受信しないとRFULLがセットされません。

RSR[1,2]の内容のデータ損失は、RSR[1,2]の第3ワードの終わりから2.5サイクル以内にDRR[1,2]がリードされれば回避できます。

次のどちらかの方法により、RFULLビットがゼロにクリアされ、後続の転送が正しくリードされます。

□ DRR[1,2]のリード

□ 受信部リセット( $\overline{\text{RRST}} = 0$ )またはデバイス・リセット

受信部を再起動するには、別のフレーム同期が必要です。

図2-28に受信オーバーラン状態を示します。シリアル・ワードBの受信完了前にシリアル・ワードAがリードされなかったため、BはまだDRR1に転送されていません。別の新しいワードCが到着し、RSR1はこのデータで一杯になります。結局DRR1はリードされますが、ワードCの終わりから2.5サイクル以内ではありません。したがって、新しいデータDによってRSR1の前のワードCが上書きされます。Dの到着後に依然としてRFULLがセットされている場合は、DRR1が時間内にリードされないと、次のワードによって上書きされます。

図2-28. シリアル・ポート受信オーバーラン

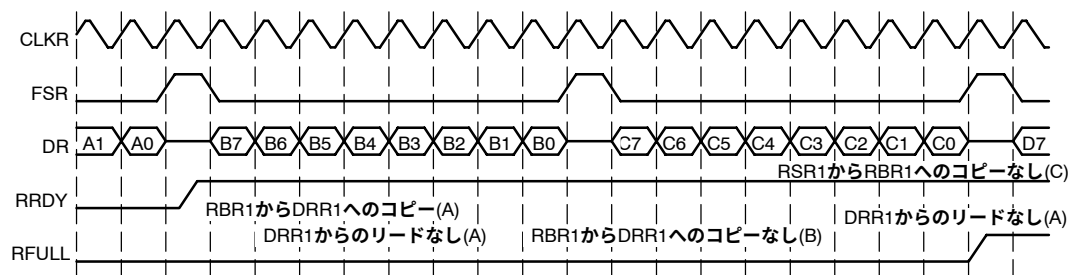
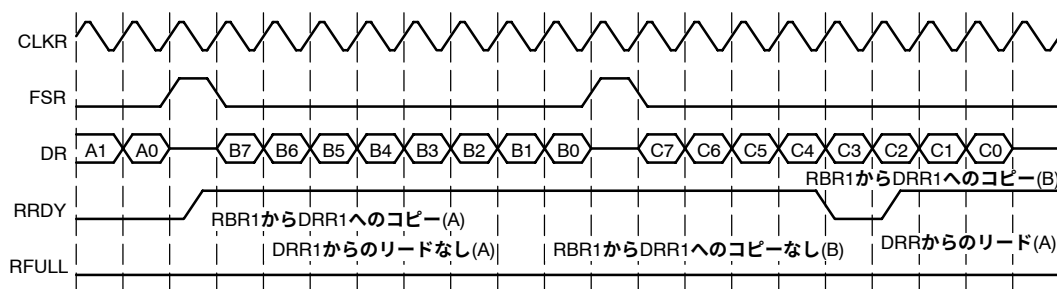


図2-29に、次のシリアル・ワードCがRSR1に完全にシフト・インされるよりも最低2.5サイクル前にDRR1の内容をリードすることで、オーバーラン状態を回避したケースを示します。この場合は、次のシリアル・ワード(C)がRSR1からRBR1に転送される前に、RBR1からDRR1へのデータBのコピーが行われます。

図2-29. シリアル・ポート受信オーバーランの回避



### 2.3.7.2 予定外の受信フレーム同期: RSYNCERR

図2-30に、受信部におけるすべての着信フレーム同期信号の処理のフローチャートを示します。この図では、受信部が起動している( $\overline{\text{RRST}} = 1$ )と仮定します。予定外のフレーム同期パルスは、外部のソースまたは内部のサンプル・レート生成器から発生します。予定外のフレーム同期パルスは、前のフレームで受信された最後のビットの前に、RDATDYビット・クロックよりも早く発生した同期パルスとして定義されます。次の4つのケースが考えられます。

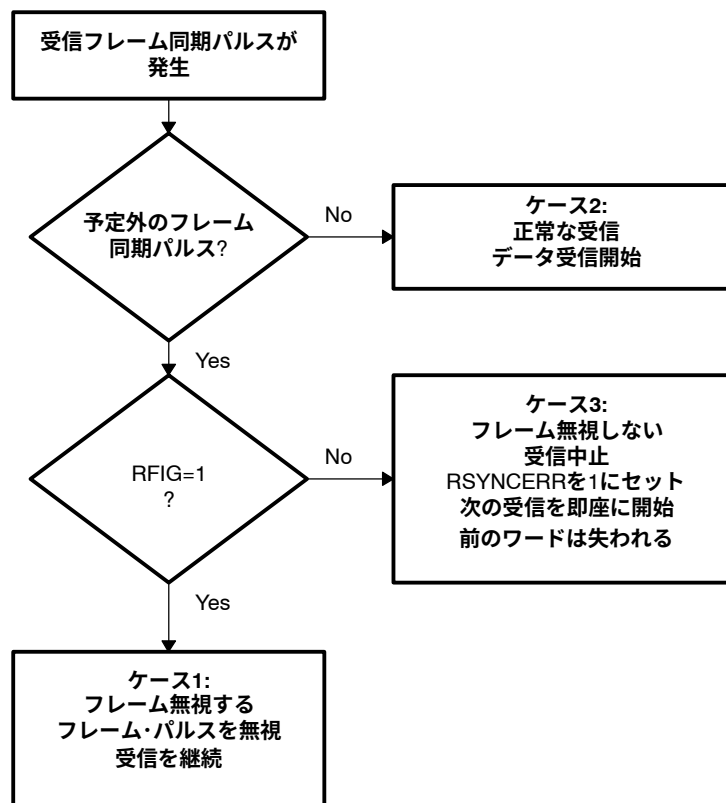
- ☐ ケース1: RFIG = 1で、予定外の内部FSRパルスが発生。このケースについては、2-42ページの第2.3.6.2節で説明し、2-43ページの図2-27に示しました。このケース1では、受信フレーム同期パルスが無視され、受信が継続されます。
- ☐ ケース2: 正常なシリアル・ポート受信。受信が進行しない場合の理由としては、次の3つが考えられます。
  - FSRが $\overline{\text{RRST}} = 1$ に設定された後の最初のFSRである。
  - FSRがRFULL状態をクリアするDRR[1,2]リード後の最初のFSRである。
  - シリアル・ポートがパケット間インターバルにある。プログラムされた受信データ遅延(RDATDLY)によって、このパケット間インターバル中に、次の受信ワードの先頭ビットより前にフレーム同期が発生する事があります。したがって、最大フレーム周波数では、フレーム同期が対応する関連ワードの先頭ビットよりもRDATDLYビット・クロック前に受信されます。

ケース2では、予定外のフレーム同期パルスではないため、受信は正常に継続されます。



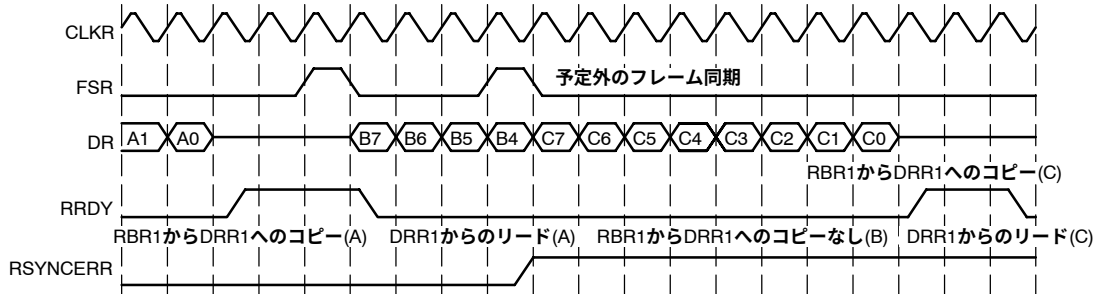
- ケース3: RFIG = 0(予定外のフレームが無視されない)で予定外の受信フレーム同期。  
 2-43ページの図2-26に、最大フレーム周波数でのこのケースを示しました。2-48ページの図2-31は、パケット間に時間間隔があり、シリアル・ポートが正常に動作しているときのこのケースを示します。予定外のフレーム同期パルスは、前のワードの最終ビットがDRピン上で受信されるよりも(RDATDLY-1)ビット・クロック前、あるいはそれより早く発生した場合に検出されます。どちらの場合も、SPCR1のRSYNCERRビットが1にセットされます。RSYNCERR<sup>†</sup>は、受信部リセット、あるいはユーザーがSPCR1のこのビットに0をライトすることによってのみクリアできます。SPCR1でRINTM = 11bの場合、RSYNCERRはCPUへの受信割り込み(RINT)を発生させます。

図2-30. 受信フレーム同期パルスへの応答



<sup>†</sup> SPCR1のRSYNCERRビットは、リード/ライト・ビットです。したがって、このビットに1を書き込むとエラー状態になります。一般には0をライトします。

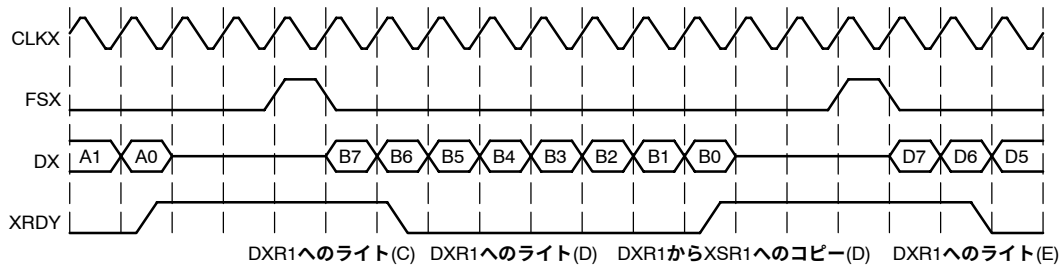
図2-31. 予定外の受信同期パルス



### 2.3.7.3 データ上書きをとまなう送信

図2-32は、DXR1のデータが送信される前に上書きされた様子を示した図です。まず、プログラマはDXR1にデータCをロードしました。その後、CがXSR1にコピーされる前にDをDXR1にライトすると、Cは上書きされます。したがって、CがDXから送信されることはありません。CPUは、DXR1にライトされる前にXRDYビットをポーリングすることにより、あるいはXRDYビットによって割り込みXINTを発生させるようにプログラム(XINTM = 00b)することによって、データの上書きを回避できます。DMAは、データ転送をXEVTと同期することで上書きを回避できます。

図2-32. データ上書きをとまなう送信



### 2.3.7.4 送信エンプティ: $\overline{\text{XEMPTY}}$

$\overline{\text{XEMPTY}}$ は、送信部でアンダーフローが発生していることを示します。次の条件のいずれによっても、 $\overline{\text{XEMPTY}}$ がアクティブになります( $\overline{\text{XEMPTY}} = 0$ )。

- 1) 送信中。DXR[1,2]からXSR[1,2]への最後のコピー以降、DXR[1,2]へのロードがなく、XSR[1,2]のデータ・ワードの全ビットが既にDXピンにシフト・アウトされている。
- 2) 送信部がリセットされ( $\overline{\text{XRST}} = 0$ 、またはデバイスがリセットされた)、再起動された。

アンダーフロー状態では、送信部はCPUまたはDMAによってDXR[1,2]に新しいワードがロードされるまで、新しいフレーム同期信号がFSXに到着するたびにDXR[1,2]の古いデータを送信し続けます。DXR[1,2]の新しいワードがXSR[1,2]に転送されると、 $\overline{\text{XEMPTY}}$ がインアクティブ( $\overline{\text{XEMPTY}} = 1$ )になります。内部でフレーム同期を生成する場合は、送信部は

DXR[1,2]からXSR[1,2]へコピーされた時に1つのフレーム同期を生成します(PCRのFSXM = 1かつSRGR2のFSGM = 0)。その他の場合は、送信部は次のフレーム同期を待ちます。

送信部のリセットが解除されると( $\overline{\text{XRST}} = 1$ )、送信レディー(XRDY = 1)および送信エンプティ( $\overline{\text{XEMPTY}} = 0$ )状態になります。内部FSXがアクティブ(ハイ)になる前にCPUまたはDMAによってDXR[1,2]にデータがロードされると、DXR[1,2]からXSR[1,2]への有効な転送が行われます。これによって、送信フレーム同期パルスが生成または検出される前であっても、最初のフレームの先頭ワードが有効になります。一方、DXRへのロードの前に送信フレーム同期が検出されると、DXからゼロが出力されます。

図2-33に、送信アンダーフローが発生している様子を示します。Bが送信された後、プログラマが後続のフレーム同期より前にDXR[1,2]へのロードを行いませんでした。したがって、DXからBが再送信されます。図2-34に、送信アンダーフロー状態が発生する直前にDXR1へのライトが行われたケースを示します。Bが送信された後、次の送信フレーム同期パルスが発生する前にCがDXR1にライトされたため、DXからCが正しく送信され、送信エンプティ状態が回避されています。

図2-33. 送信エンプティ

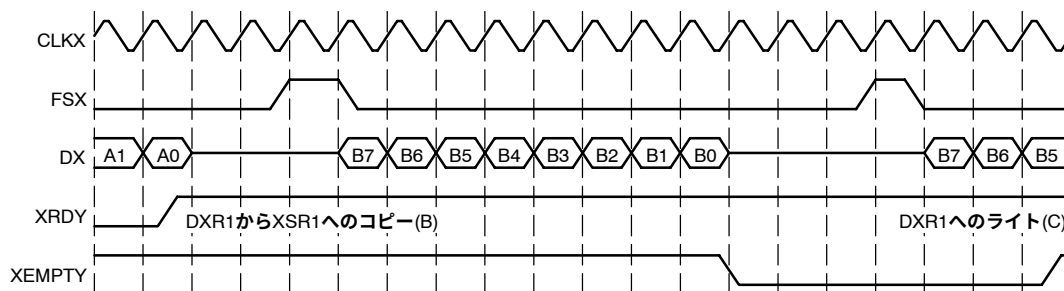
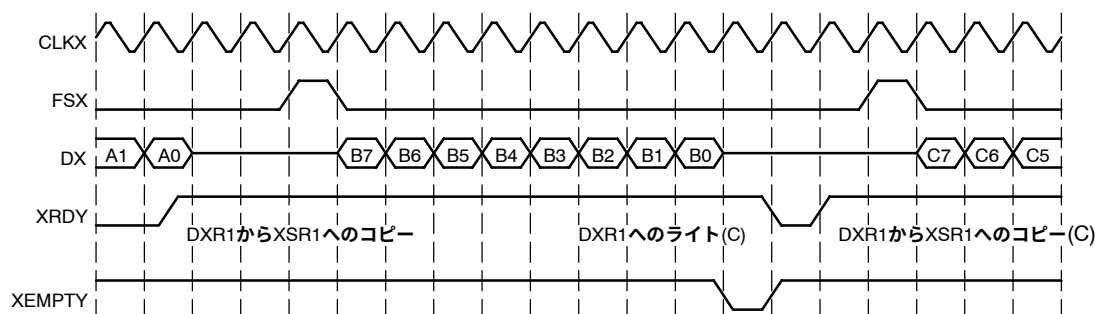


図2-34. 送信エンプティの回避

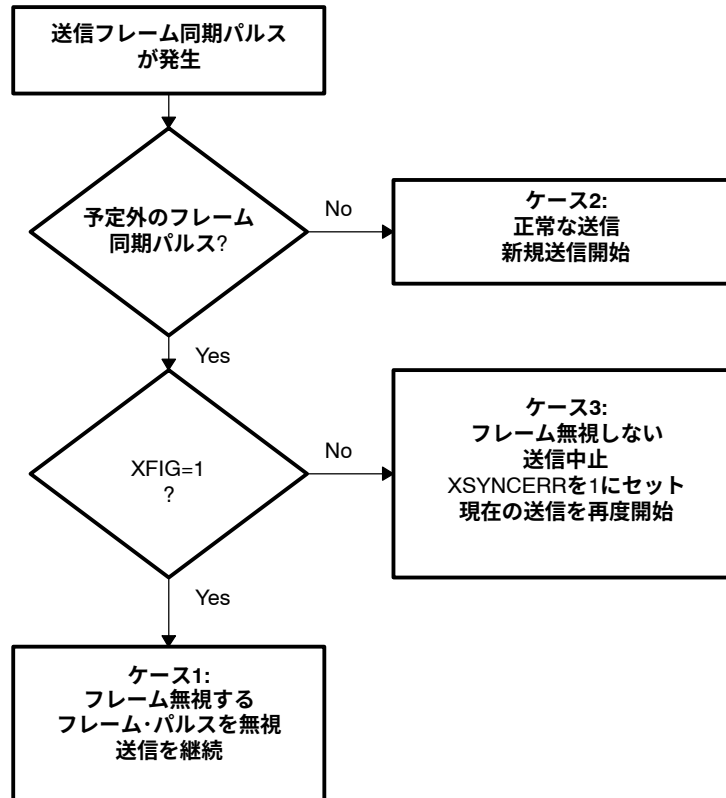


## 2.3.7.5 予定外の送信フレーム同期: XSYNCERR

2-51ページの図2-35に、送信部におけるすべての着信フレーム同期信号の処理のフローチャートを示します。この図では、送信部が起動している( $\overline{\text{XRST}} = 1$ )と仮定しています。予定外の送信フレーム同期パルスは、前のフレームで送信された最後のビットの前にXDADLYビット・クロックより早く発生した同期パルスとして定義されます。次の3つのケースが考えられます。

- ケース1: XFIG = 1で予定外のFSXパルスが発生。このケースについては、2-42ページの第2.3.6.2節で説明し、2-43ページの図2-27に示しました。
- ケース2: 正常なシリアル・ポート送信。このケースについては、2-39ページの第2.3.5.2節で説明しています。送信が進行しない場合、理由としては次の2つが考えられます。
  - このFSXパルスが $\overline{\text{XRST}} = 1$ 後の最初のFSXである。
  - シリアル・ポートがパケット間インターバルにある。プログラムされたデータ遅延(XDATDLY)により、このパケット間インターバル中に、次のワードの先頭ビットが送信される前にフレーム同期が発生することがあります。したがって、最大パケット周波数では、フレーム同期が対応するワードの先頭ビットよりもXDATDLYビット・クロック前に送信されることがあります。

図2-35. 送信フレーム同期への応答

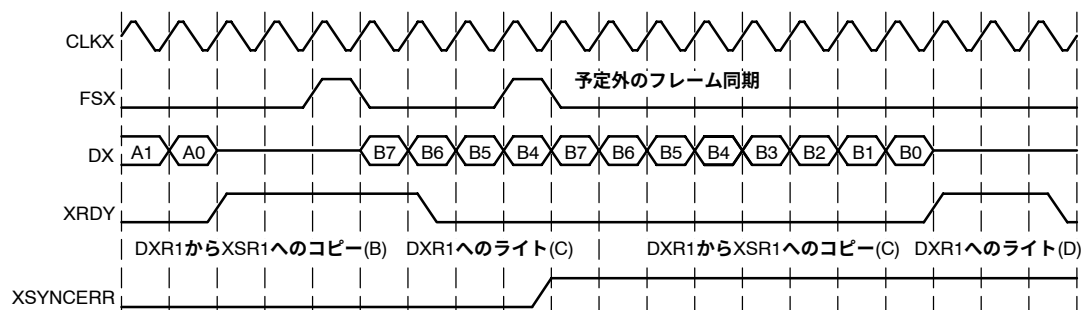


- ケース3: XFIG = 0で、予定外の送信フレーム同期。図2-26に、最大フレーム周波数で XFIG = 0の場合における後続フレーム同期のケースを示しました。図2-36は、パケット間にインターバルがあり、シリアル・ポートが正常に動作しているときのこのケースを示します。どちらの場合も、SPCR2のXSYNCERRビットが1にセットされます。XSYNCERRは、送信部リセット、あるいはユーザーがSPCR2のこのビットに0をライトすることによってのみクリアできます。SPCR2のXINTM = 11bの場合、XSYNCERRはCPUへの送信割り込み(XINT)を発生させます。

**注：**

SPCR2のXSYNCERRビットは、リード/ライト・ビットです。したがって、このビットに1を書き込むとエラー状態になります。一般には0をライトします。

図2-36. 予定外の送信フレーム同期パルス



### 2.3.8 受信データの位置調整と符号拡張: RJUST

SPCR1のRJUSTフィールドは、RBR[1,2]のデータをDRR[1,2]で(MSBに対して)右詰めまたは左詰めのどちらにするかを選択します。右詰めを選択した場合、RJUSTでさらに、データを符号拡張するか、ゼロ詰めするかを選択します。表2-15に、12ビットの受信データ値0xABCに対するRJUSTでの各モードの結果を示します。表2-16には、20ビットの受信データ値0xABCDEに対するRJUSTの各モードでの結果を示します。

表2-15. 12ビットのデータ0xABCでのRJUSTフィールドの使用

| RJUST | 位置調整 | 拡張       | DRR2の値 | DRR1の値 |
|-------|------|----------|--------|--------|
| 00    | 右    | MSBのゼロ詰め | 0x0000 | 0x0ABC |
| 01    | 右    | MSBの符号拡張 | 0xFFFF | 0xFABC |
| 10    | 左    | LSBのゼロ詰め | 0x0000 | 0xABC0 |
| 11    | 予約   | 予約       | 予約     | 予約     |

表2-16. 20ビットのデータ0xABCDEでのRJUSTフィールドの使用

| RJUST | 位置調整 | 拡張       | DRR2の値 | DRR1の値 |
|-------|------|----------|--------|--------|
| 00    | 右    | MSBのゼロ詰め | 0x000A | 0xBCDE |
| 01    | 右    | MSBの符号拡張 | 0xFFFA | 0xBCDE |
| 10    | 左    | LSBのゼロ詰め | 0xABCD | 0xE000 |
| 11    | 予約   | 予約       | 予約     | 予約     |

## 2.4 μ 則/A 則圧伸ハードウェアの動作: (R/X)COMPAND

圧伸(圧縮および伸張)ハードウェアでは、μ 則フォーマットまたはA 則フォーマットでのデータの圧縮および伸張ができます。米国および日本で採用されている圧伸規格はμ 則です。欧州ではA 則が採用されています。μ 則およびA 則のログPCMの仕様は、ITU-T(旧CCITT) G.711勧告に規定されています。A 則とμ 則では、それぞれ13ビットおよび14ビットのダイナミック・レンジを使用できます。この範囲外の値は、最も大きな正の値または最も大きな負の値として扱われます。したがって、このような圧伸を最大限に機能させるには、CPUまたはDMAを通じたMcBSPとのデータ転送において、またはMcBSPから送信されるデータが、少なくとも16ビットの精度が必要になります。

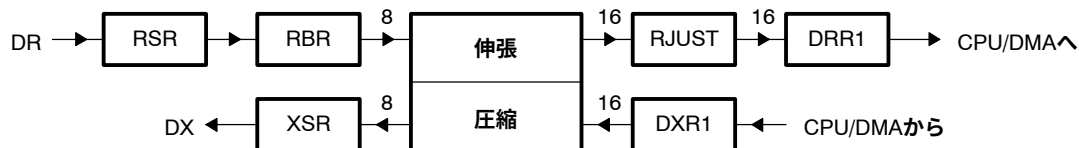
μ 則フォーマットとA 則フォーマットは、データを8ビットのワードにエンコードします。圧伸されるデータは常に8ビット幅です。したがって、対応する(R/X)WDLEN[1,2]を、8ビット幅のシリアル・データ・ストリームを示す0に設定する必要があります。圧伸がイネーブルであり、フレームのフェーズのどちらかで8ビットのワード長が設定されていない場合は、ワード長が8ビットであるかのように圧伸が行われます。

圧伸をすると、送信データは指定された圧伸方法に従ってエンコードされ、受信データは2の補数のフォーマットでデコードされます。次の各表中の(R/X)CR2の(R/X)COMPANDを適切に設定して希望のフォーマットを選択します。

- ☐ 2-16ページの表2-7「受信制御レジスタ1(RCR1)のビット/フィールドの説明」
- ☐ 2-17ページの表2-8「受信制御レジスタ2(RCR2)のビット/フィールドの説明」
- ☐ 2-19ページの表2-9「送信制御レジスタ1(XCR1)のビット/フィールドの説明」
- ☐ 2-20ページの表2-10「送信制御レジスタ2(XCR2)のビット/フィールドの説明」

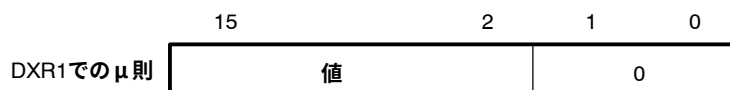
圧伸は、図2-37に示すように、DXR1からXSR1への、およびRBR1からDRR1へのデータのコピー中に行われます。

図2-37. 圧伸フロー



送信データを正しく圧縮するには、データをDXR1に左詰めにしてライトします。μ 則を使用すると、図2-38に示すように、レジスタで14のデータ・ビットが左詰めされ、残りの下位2ビットがゼロ詰めされます。

図2-38. μ 則送信データ圧伸フォーマット



A 則を使用すると、図2-39に示すように、レジスタで13データ・ビットが左詰めされ、残りの下位3ビットがゼロ詰めされます。

図2-39. A 則送信データ圧伸フォーマット



受信の場合は、RBR1の8ビット圧縮データが、DRR1で左詰め16ビット・データに伸張されます。圧伸を使用するとRJUSTは無視されます。



## 2.4.1 内部データの圧伸

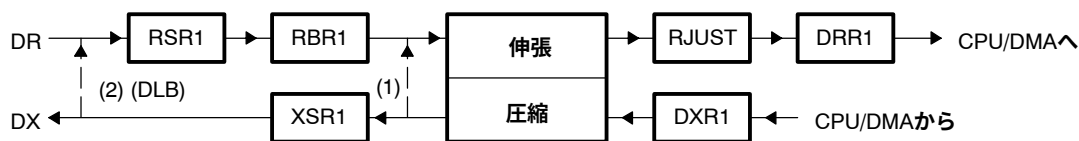
McBSPが他に使用されていない(シリアル・ポートの送受信部分がリセットされている)場合でも、圧伸ハードウェアは内部データを圧伸できます。これは、次のように使用できます。

- ☐ リニア・フォーマットをμ則フォーマットまたはA則フォーマットに変換する。
- ☐ μ則フォーマットまたはA則フォーマットをリニア・フォーマットに変換する。
- ☐ リニア・データを送信し、このデータを圧縮および再伸張することで、圧伸における量子化効果を評価する。これは、XCOMPANDとRCOMPANDの両方で同じ圧伸フォーマットをイネーブルにしている場合にのみ使用することができます。

図2-40に、McBSPが内部データを圧伸できる2つの方法を示します。これら2つの方法に使われるデータ・パスが、次のことを示すために使用されています。

- 1) シリアル・ポートの送信部分と受信部分の両方がリセットされると、DRR1とDXR1は圧伸ロジックによって内部で接続されます。DXR1の値はXCOMPANDの設定どおりに圧縮され、その後RCOMPANDの設定どおりに伸張されます。RRDYビットとXRDYビットはどちらもセットされないことに注意してください。しかし、データはDXR1にライトされてから4CPUクロック以内にDRR1で使用可能になります。この方法の利点は高速であるということです。欠点として、CPUおよびDMAに対してデータの流れを制御するための同期を発生することができません。(X/R)COMPANDビットが1xb、つまりA則またはμ則による圧伸に設定されている場合、DRR1とDXR1は内部で接続されることに注意してください。
- 2) RCOMPANDとXCOMPANDによって圧伸をイネーブルにして、デジタル・ループ・バック・モードをイネーブルにする。受信割り込みおよび送信割り込み(RINTM = 0の場合のRINT、XINTM = 0の場合のXINT)または同期イベント(REVTおよびXEVT)は、CPUまたはDMAがそれぞれの変換に同期をとることができます。この場合圧伸にかかる時間は、選択されたシリアル・ビット・レートによって決まります。

図2-40. 内部データの圧伸



#### 2.4.1.1 ビット・オーダリング

2

通常、McBSP上のすべての転送は、MSBから送受信されます。しかし、8ビットのデータ・プロトコル(圧伸データを使用しないもの)によっては、LSBから転送する必要があります。(R/X)CR2で(R/X)COMPAND = 01bに設定することで、8ビット・ワードのビット順がシリアル・ポートに送信される前に反転します(LSBからになります)。この機能は圧伸と同様に、対応する(R/X)WDLEN[1,2]が0に設定され、シリアル転送のワード長として8ビットを選択した場合のみイネーブルになります。フレームのどちらかのフェーズに8ビットのワード長が設定されていない場合、McBSPはワード長が8ビットであると仮定し、LSBからのビット順が使用されます。

McBSPには、受信部および送信部の両方について、クロック生成/同期とフレーム生成/同期の手段をいくつか選択できます。サンプル・レート生成器は、クロックとフレーム同期を、両方の部分に供給できます。外部クロックまたはフレーム同期を、送受信部分で個別に選択できます。図2-41に、クロックおよびフレーム選択回路のブロック図を示します。この各機能については、以後各節で説明します。

[illegible]

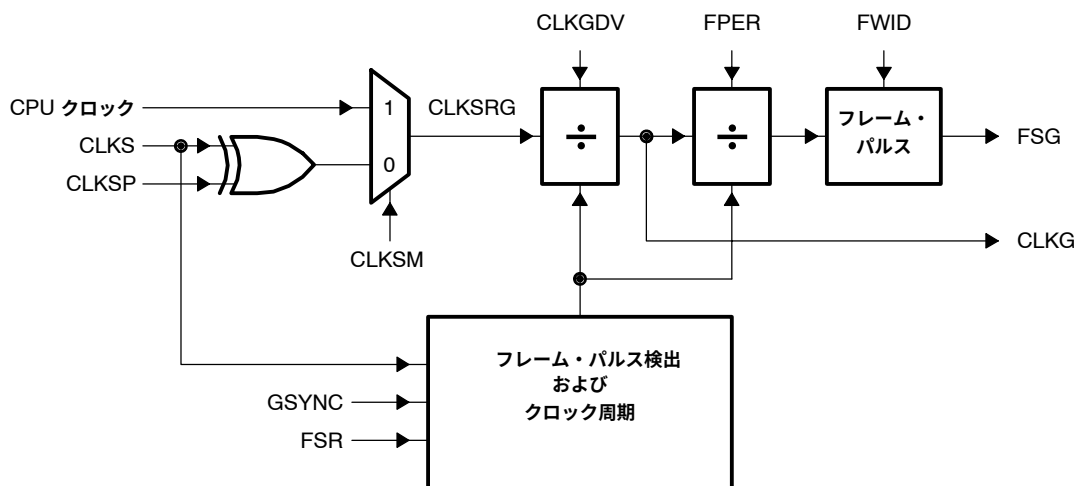
### 2.5.1 サンプル・レート生成器のクロックとフレームの生成

サンプル・レート生成器は、図2-42に示すように、プログラマブルなデータ・クロック (CLKG) およびフレーム信号 (FSG) を生成する3段のクロック分周器で構成されます。CLKG とFSGは、受信または送信のクロック (CLKR/X) およびフレーム同期 (FSR/X) を生成するプログラム可能なMcBSP内部信号です。サンプル・レート生成器は、内部クロック・ソースによって、あるいは外部クロック・ソースから供給される内部クロックによって駆動されるようにプログラムできます。サンプル・レート生成器の回路は3つのステージで、次の計算を行います。

- クロック分周 (CLKGDV): データ・ビット・クロック当たりの入力クロック数
- フレーム周期分周 (FPER): データ・ビット・クロック数で表したフレーム周期
- フレーム幅 (FWID): データ・ビット・クロックで表されたアクティブ・フレーム・パルスの幅

さらに、フレーム・パルス検出およびクロック同期モジュールにより、クロック分周を着信フレーム・パルスと同期させることができます。デバイス・リセット中のサンプル・レート生成器の動作については、2-22ページの第2.3.1節「シリアル・ポートのリセット: (R/X) $\overline{\text{RST}}$ 、および  $\overline{\text{RESET}}$ 」で説明しています。

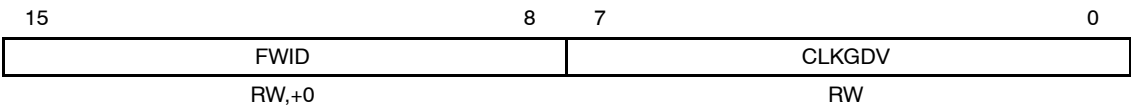
図2-42. サンプル・レート生成器



2.5.1.1 サンプル・レート生成器レジスタ(SRGR[1,2])

サンプル・レート生成器レジスタ1および2(SRGR[1,2])は、サンプル・レート生成器の各機能の動作を制御します。これらのレジスタとそのビット/フィールドの説明は、それぞれ図2-43と表2-17、および図2-44と表2-18に示します。これらの図および表に続く各節では、SRGR[1,2]ビット/フィールドを使ってSRGRの動作を設定する方法を説明します。

図2-43. サンプル・レート生成器レジスタ1(SRGR1)



注: R = リード、W = ライト、+0 = リセット時の値

表2-17. サンプル・レート生成器レジスタ1(SRGR1)のビット/フィールドの説明

| ビット  | 名前     | 機能                                                                                | 節       |
|------|--------|-----------------------------------------------------------------------------------|---------|
| 15-8 | FWID   | フレーム幅。(このフィールドの値+1)により、フレーム同期パルスFSGのアクティブ期間中の幅が決まります。<br>範囲: 1~256のCLKG期間         | 2.5.3.1 |
| 7-0  | CLKGDV | サンプル・レート生成器クロック分周器<br>この値は、希望するサンプル・レート生成器クロック周波数を生成するための分周率として使用されます。デフォルト値は1です。 | 2.5.2.2 |

図2-44. サンプル・レート生成器レジスタ2(SRGR2)

| 15    | 14    | 13    | 12    | 11    | 0 |
|-------|-------|-------|-------|-------|---|
| GSYNC | CLKSP | CLKSM | FSGM  | FPER  |   |
| RW,+0 | RW,+0 | RW    | RW,+0 | RW,+0 |   |

注: R = リード、W = ライト、+0 = リセット時の値

表2-18. サンプル・レート生成器レジスタ2(SRGR2)のビット/フィールドの説明

| ビット | 名前    | 機能                                                                                                                                                                                                                                                                                               | 節       |
|-----|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| 15  | GSYNC | <p><b>サンプル・レート生成器のクロック同期</b></p> <p>外部クロック(CLKS)がサンプル・レート生成器クロックを駆動する場合 (CLKSM = 0)にのみ使用されます。</p> <p>GSYNC = 0    サンプル・レート生成器クロック(CLKG)がフリー・ラン</p> <p>GSYNC = 1    サンプル・レート生成器が動作中。しかし、受信フレーム同期信号(FSR)が検出された時のみCLKGが再同期され、フレーム同期信号(FSG)が生成されます。また、フレーム周期は外部フレーム同期パルスにより決定されるので、FPERは参照されません。</p> | 2.5.2.4 |
| 14  | CLKSP | <p><b>CLKS極性クロック・エッジ選択</b></p> <p>外部クロックCLKSがサンプル・レート生成器クロックを駆動する場合 (CLKSM = 0)にのみ使用されます。</p> <p>CLKSP = 0    CLKSの立ち上がりエッジでCLKGおよびFSGが生成されます。</p> <p>CLKSP = 1    CLKSの立ち下がりエッジでCLKGおよびFSGが生成されます。</p>                                                                                          | 2.5.2.3 |
| 13  | CLKSM | <p><b>McBSPサンプル・レート生成器クロック・モード</b></p> <p>CLKSM = 0    サンプル・レート生成器クロックがCLKSピンから供給されます。</p> <p>CLKSM = 1    サンプル・レート生成器クロックがCPUクロックから供給されます。</p>                                                                                                                                                  | 2.5.2.2 |

表2-18. サンプル・レート生成器レジスタ2(SRGR2)のビット/フィールドの説明(続き)

| ビット  | 名前   | 機能                                                                                                                                                                                                                             | 節       |
|------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| 12   | FSGM | <p>サンプル・レート生成器の送信フレーム同期モード</p> <p>PCRでFSXM = 1の場合に使用されます。</p> <p>FSGM = 0     DXR[1,2]からXSR[1,2]へのコピーによりフレーム同期信号(FSX)を生成します。FSGM = 0の場合は、FPERとFWIDが無視されます。</p> <p>FSGM = 1     サンプル・レート生成器のフレーム同期信号FSGにより送信フレーム同期信号を生成します。</p> | 2.5.3.3 |
| 11-0 | FPER | <p>フレーム周期。(このフィールドの値+1)により、次のフレーム同期信号がいつアクティブになるかが決まります。</p> <p>範囲: 1~4096 CLKG期間</p>                                                                                                                                          | 2.5.3.1 |

### 2.5.1.2 サンプル・レート生成器のリセット手順

サンプル・レート生成器のリセットおよび初期化手順は、次のとおりです。

- 1) デバイス・リセット中の場合は $\overline{\text{GRST}} = 0$ です。その他の正常動作中は、CLKGまたはFSGがMcBSPのどの部分でも使用されていないければ、SPCR2で $\overline{\text{GRST}} = 0$ に設定するとサンプル・レート生成器をリセットできます。デバイス・リセットによる $\overline{\text{GRST}} = 0$ の場合は、CLKGはCPUクロックの2分周で生成され、FSGはインアクティブ(ロー)になります。ユーザーによって $\overline{\text{GRST}} = 0$ にプログラムした場合は、CLKGおよびFSGはインアクティブ(ロー)になります。必要であれば、(R/X) $\overline{\text{RST}} = 0$ に設定します。
- 2) 必要に応じてSRGR[1,2]をプログラムします。必要であれば、他の制御レジスタに目的の値をライトします。ただし、それぞれの部分(送受信部)がリセット状態である場合に限ります。
- 3) 2 CLKSRGクロック待ちます。これにより、内部で正しい同期が行われます。
- 4)  $\overline{\text{GRST}} = 1$ に設定してサンプル・レート生成器をイネーブルにします。
- 5) 2 CLKGビット・クロック待ちます。
- 6) 必要な場合は、受信部または送信部のリセットを解除します((R/X) $\overline{\text{RST}} = 1$ )。
- 7) CLKSRGの次の立ち上がりエッジで、CLKGが1になり、CLKSM = 1の場合は(CPUクロック)/(1+CLKGDV)に、CLKSM = 0の場合は(CLKSクロック)/(1+CLKGDV)に等しい周波数でクロック生成を開始します。
- 8) DXR[1,2]にデータをロードする等データ転送に必要なセット・アップが終了した後、内部生成フレーム同期パルスが必要であれば $\overline{\text{FRST}}$ に1をライトすることができます。FSGは、プログラムされた値の8 CLKGクロックが経過した後にアクティブ・ハイ・エッジで生成されます。

## 2.5.2 データ・クロックの生成

受信/送信クロック・モードが1に設定されている( $\text{CLK[R/X]M} = 1$ )場合、データ・クロック( $\text{CLK[R/X]}$ )は内部のサンプル・レート生成器出力クロックCLKGによって駆動されます。受信部および送信部について個別に、様々な種類のデータ・ビット・クロックの中から選択できます。これには、次の選択肢があります。

- サンプル・レート生成器への入力クロックは、CPUクロックまたは専用の外部クロック入力(CLKS)のどちらかを選択できます。
- サンプル・レート生成器の入力クロック(CPUクロックまたは外部クロックCLKS)ソースを、プログラマブルな値( $\text{CLKGDV}$ )で分周することによりCLKGを生成することができます。サンプル・レート生成器へのソースに関係なく、CLKSRG(図2-42参照)の立ち上がりエッジでCLKGおよびFSGが生成されます(第2.5.2.3節も参照のこと)。

### 2.5.2.1 入力クロック・ソース・モード: CLKSM

SRGR2のCLKSMビットは、サンプル・レート生成器入力クロックのソースとして、CPUクロック( $\text{CLKSM} = 1$ )または外部クロック入力( $\text{CLKSM} = 0$ )CLKSのどちらかを選択します。分周率はサンプル・レート生成器によって、カウント・ダウンされるカウント数を示し、この選択された入力クロックと同期します。 $\text{CLKSM} = 1$ の場合は、 $\text{CLKGDV}$ の最小値は1となります。

### 2.5.2.2 サンプル・レート生成器のデータ・ビット・クロック・レート: CLKGDV

最初の分周ステージは、入力クロックからシリアル・データ・ビット・クロックを生成します。この分周ステージでは、分周カウント数を保持する $\text{CLKGDV}$ によって前もってロードされるカウンタを使用します。このステージの出力は、サンプル・レート生成器出力CLKGから出力されるデータ・ビット・クロックであり、第2、第3の分周ステージの入力となります。

CLKGの周波数は、サンプル・レート生成器入力クロックの $1/(\text{CLKGDV} + 1)$ に等しくなります。したがって、サンプル・レート生成器入力クロックの周波数は1から256の間の値で分周されます。 $\text{CLKGDV}$ が奇数または0の場合、CLKGのデューティ・サイクルは50%です。 $\text{CLKGDV}$ が偶数値 $2p$ であり、奇数の分周カウント数を示す場合は、ハイ状態の持続時間は $p + 1$ サイクルであり、ロー状態の持続時間は $p$ サイクルです。

### 2.5.2.3 ビット・クロック極性: CLKSP

$\text{CLKSM} = 0$ に設定すると外部クロック(CLKS)により、サンプル・レート生成器クロック分周器を駆動させることができます。この場合SRGR2のCLKSPビットは、サンプル・レート生成器のデータ・ビット・クロック(CLKG)およびフレーム同期信号(FSG)が生成される時のCLKSのエッジを選択します。CLKSRGの立ち上がりエッジ(図2-42参照)でCLKGおよびFSGが生成されるため、 $\text{CLKSP} = 0$ のときはCLKSの立ち上がりエッジ、 $\text{CLKSP} = 1$ のときはCLKSの立ち下がりエッジにより、データ・ビット・レート・クロック(CLKG)とフレーム同期(FSG)が遷移します。



## 2.5.2.4 ビット・クロックとフレーム同期

CLKSがサンプル・レート生成器を駆動するよう設定した( $CLKSM = 0$ )場合、 $GSYNC$ により、CLKSに対するCLKGのタイミングを設定できます。 $GSYNC = 1$ に設定すると、McBSPと、McBSPが通信する外部デバイスが、同じ位相関係でCLKSを分周します。 $GSYNC = 0$ の場合はこの機能がディセーブルされ、CLKGは独自に動作し、再同期されません。 $GSYNC = 1$ の場合、FSRのインアクティブからアクティブへの遷移により、CLKGの再同期とFSGの生成が行われます。CLKGは同期後、常にハイ状態で開始されます。また、CLKGは、FSRパルスの長さにかかわらず常に、CLKGを生成するCLKSのエッジと同じエッジで検出されます。外部FSRが供給されていても、 $GSYNC = 1$ であればFSGは内部受信フレーム同期を生成できます。 $GSYNC = 1$ の場合、フレーム周期は外部フレーム同期パルスの到着によって決まるため、FPERは参照されません。

図2-45と図2-46に、CLKRおよびFSRの各極性でのビット・クロックとフレーム同期の動作を示します。これらの図では、FSGの幅は1 CLKGの幅となるように、FWID = 0と仮定します。

図2-45.  $GSYNC = 1$ かつ $CLKGDV = 1$ の場合のCLKG同期およびFSG生成

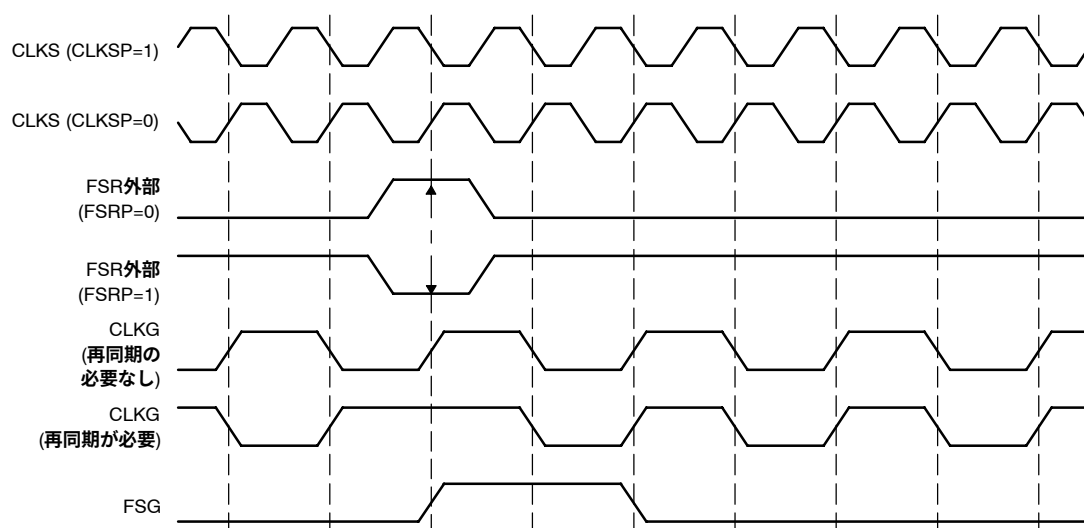


図2-46. GSYNC = 1かつCLKGDV = 3の場合のCLKG同期およびFSG生成

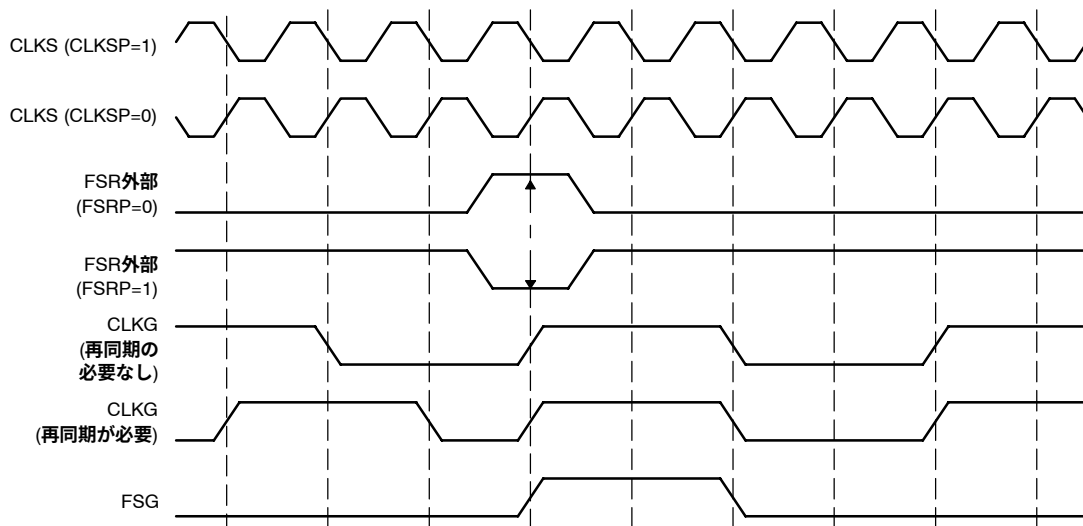


図2-45と図2-46は、GSYNC = 1でありCLKGが最初から同期されている場合と、GSYNC = 1でありCLKGが最初に同期されていない場合のCLKGの動作を示しています。

GSYNC = 1の場合、次の条件により、送信部が受信部と同期して動作できます。

- 1) FSXがサンプル・レート生成器のフレーム同期FSGによって駆動されるようプログラムされている(SRGMでFSGM = 1かつPCRでFSXM = 1)。入力FSRをCLKGの立ち下がりエッジでサンプリングできるようなタイミングであれば、PCRでFSXM = 0に設定し、外部でFSRをFSXに接続することでFSRを代用できます。
- 2) サンプル・レート生成器クロックが送信ビット・クロックおよび受信ビット・クロックを駆動する(SPCR[1,2]でCLK[R/X]M = 1)。したがって、CLK(R/X)ピンを他のソースでドライブすることはできません。

#### 2.5.2.5 デジタル・ループ・バック・モード: DLB

SPCR1でDLB = 1に設定すると、デジタル・ループ・バック・モードがイネーブルになります。DLBモードでは、図2-41に示したように、DR、FSR、およびCLKRがマルチプレクサを通じてそれぞれDX、FSX、およびCLKXに内部で接続されます。DLBモードでは、DSPデバイス単体でシリアル・ポートのプログラムをテストできます。図2-42に、デジタル・ループ・バック・モードでの受信制御入力のブロック図を示しています。

## 2.5.2.6 受信クロックの選択: DLB、CLKRM

表2-19に、デジタル・ループ・バック・ビット(DLB)とPCRのCLKRMビットによって受信部クロックを選択する方法を示します。デジタル・ループ・バック・モード(DLB = 1)では、送信クロックが受信部を駆動します。CLKRMは、CLKRピンを入力または出力に設定します。

表2-19. 受信クロックの選択

| SPCR1<br>のDLB | PCRの<br>CLKRM | 受信クロックのソース                                                        | CLKRピン                                     |
|---------------|---------------|-------------------------------------------------------------------|--------------------------------------------|
| 0             | 0             | CLKRピンは入力であり、外部クロックによってドライブされます。また、CLKRPの設定によって極性を反転して用いることができます。 | 入力                                         |
| 0             | 1             | サンプル・レート生成器クロック(CLKG)がCLKRを駆動します。                                 | 出力。CLKRPの設定によってCLKGは反転して出力することができます。       |
| 1             | 0             | 内部CLKXが設定どおりに受信クロック内部CLKRを駆動し、表2-20.に示すように反転できます。                 | ハイ・インピーダンス                                 |
| 1             | 1             | 内部CLKXが設定どおりに内部CLKRを駆動し、表2-20に示すように反転できます。                        | 出力。CLKR(送信と同様に)は送出される前にCLKRPの設定によって反転できます。 |

## 2.5.2.7 送信クロックの選択: CLKXM

表2-20に、PCRのCLKXMビットによって送信クロックを選択し、CLKXピンを入力または出力に設定する方法を示します。

表2-20. 送信クロックの選択

| PCRの<br>CLKXM | 送信クロックのソース                                               | CLKXピン                                   |
|---------------|----------------------------------------------------------|------------------------------------------|
| 0             | 外部クロックがCLKX入力ピンをドライブします。CLKXはCLKXPの設定によって反転して用いることができます。 | 入力                                       |
| 1             | サンプル・レート生成器クロックCLKGが送信クロックを駆動します。                        | 出力。CLKGはCLKXから送出される前にCLKXPの設定によって反転できます。 |

### 2.5.3 フレーム同期信号の生成

データ・フレーム同期は、データ・ビット・クロックと同様、全てのデータ遅延に対して、受信部と送信部で個別にプログラムできます。SPCR2の $\overline{\text{FRST}}$ ビットを1に設定すると、フレーム同期生成ロジックがアクティブになり、SRGR2でFSGM = 1であればフレーム同期信号を生成します。フレーム同期のプログラミングには、次のような設定ができます。

- ☐ フレーム・パルスの周期とアクティブ幅を、サンプル・レート生成器レジスタ(SRGR1)で設定できます。
- ☐ 送信部は、DXR[1,2]からXSR[1,2]へのコピーによってフレーム同期信号を生成できます。これは、DXR1からXSR1へのコピーが行われるたびにフレーム同期を生成することを意味します。必要に応じてデータ遅延をプログラムできます。しかしこの方法では、データ遅延が1および2の場合に最大フレーム周波数で動作させることはできません。この制限は、フレーム無視ビット(R/X)FIG = 1に設定することで解決できます。
- ☐ 受信部分と送信部分において、それぞれFSRピンとFSXピンで外部フレーム同期を個別に選択できます。

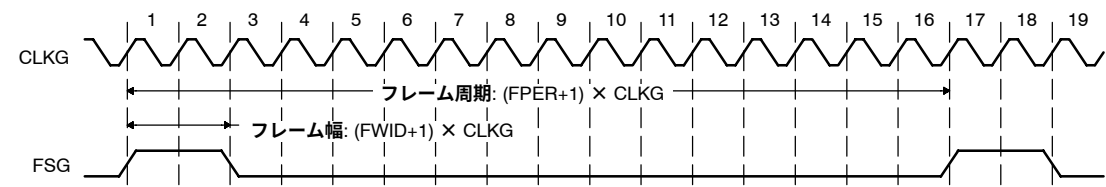
#### 2.5.3.1 フレーム周期とフレーム幅: FPERおよびFWID

FPERとFWIDは、ダウン・カウンタとして機能します。FPERステージは、生成されたデータ・クロックを4095から0までカウント・ダウンする12ビットのダウン・カウンタです。FPERはフレーム同期パルスがアクティブになる周期を制御します。サンプル・レート生成器のFWIDステージは、8ビットのダウン・カウンタです。FWIDフィールドはフレーム同期パルスのアクティブ幅を制御します。これらのカウンタには、それぞれFPERおよびFWIDにプログラムされた値がロードされます。

サンプル・レート生成器のリセットが解除されると、FSGはインアクティブ状態になります。その後、 $\overline{\text{FRST}} = 1$ かつFSGM = 1になると、フレーム同期が生成されます。フレーム幅の値(FWID+1)は、ゼロに達するまでCLKGサイクルごとにカウント・ダウンされ、ゼロになるとFSGがローになります。したがって(FWID + 1)の値により、フレーム・パルスのアクティブ幅が1~256データ・ビット・クロックの範囲で決定されます。同時に、フレーム周期の値(FPER+1)もカウント・ダウンされます。この値がゼロに達するとFSGがハイになり、再度新しいフレームを生成します。

通常、FWIDはWDLEN[1,2]よりも小さい値にプログラムします。そうすれば、(FPER+1)の値は1~4096データ・ビットのフレーム長を決定します。GSYNC = 1の場合は、FPERは参照されません。図2-47に、周期が16 CLKG(FPER = 15または00001111b)のフレームと、アクティブ幅が2 CLKG(FWID = 1)のフレームを示します。

図2-47. プログラム可能なフレーム周期とフレーム幅



2

2.5.3.2 受信フレーム同期の選択: DLB、FSRM、GSYNC

表2-21に、受信フレーム同期信号を供給する各種ソースの選択方法を示します。デジタル・ループバック・モード(DLB = 1)では、送信フレーム同期信号が受信フレーム同期信号として使用され、DRがDXに内部で接続されます。

表2-21. 受信フレーム同期の選択

| SPCR1<br>のDLB | PCRの<br>FSRM | SRGR2<br>の<br>GSYNC | 受信フレーム同期のソース                                                               | FSRピン                                                           |
|---------------|--------------|---------------------|----------------------------------------------------------------------------|-----------------------------------------------------------------|
| 0             | 0            | x                   | 外部のフレーム同期信号がFSR入力ピンをドライブします。その後、FSRPの設定により内部FSRとして使用される前に反転できます。           | 入力                                                              |
| 0             | 1            | 0                   | $\overline{\text{FRST}} = 1$ の時、内部FSRがサンプル・レート生成器のフレーム同期信号(FSG)によって駆動されます。 | 出力。FSGはFSRPの設定によってFSRピンから送出される前に反転できます。                         |
| 0             | 1            | 1                   | $\overline{\text{FRST}} = 1$ の時、内部FSRがサンプル・レート生成器のフレーム同期信号(FSG)によって駆動されます。 | 入力。FSRでの外部フレーム同期入力がCLKGの同期とFSGの生成に使用されます。                       |
| 1             | 0            | 0                   | 内部FSXが内部FSRを生成します。FSXは表2-22のように選択します。                                      | ハイ・インピーダンス                                                      |
| 1             | X            | 1                   | 内部FSXが内部FSRを生成します。FSXは表2-22のように選択します。                                      | 入力。外部FSRはフレーム同期には使用されませんが、GSYNC = 1であるためCLKGの同期とFSGの生成には使用されます。 |
| 1             | 1            | 0                   | 内部FSXが内部FSRを生成します。FSXは表2-22のように選択します。                                      | 出力。受信(送信と同じ)フレーム同期はFSRPの設定によって送出される前に反転できます。                    |

### 2.5.3.3 送信フレーム同期信号の選択: FSXM、FSGM

2

表2-22に、送信フレーム同期パルスのソースの選択方法を示します。次の3つの選択肢があります。

- 1) 外部フレーム同期入力
- 2) サンプル・レート生成器のフレーム同期信号FSG
- 3) DXR[1,2]からXSR[1,2]へのコピーが行われたことを示す信号

表2-22. 送信フレーム同期の選択

| PCRの<br>FSXM | SRGRの<br>FSGM | 送信フレーム同期のソース                                            | FSXピン                                                        |
|--------------|---------------|---------------------------------------------------------|--------------------------------------------------------------|
| 0            | x             | FSXピンでの外部フレーム同期入力。これは、内部FSXとして使用される前にFSXPの設定によって反転できます。 | 入力                                                           |
| 1            | 1             | FRST = 1の時、サンプル・レート生成器のフレーム同期信号(FSG)が内部FSXを生成。          | 出力。FSGはFSXピンから送出される前にFSXPの設定によって反転します。                       |
| 1            | 0             | DXR[1,2]からXSR[1,2]へのコピーにより、送信フレーム同期信号がアクティブになります。       | 出力。1ビット・クロック幅の信号が、FSXピンから送出されます。信号は出力される前にFSXPの設定によって反転できます。 |

### 2.5.3.4 初期化のためのフレーム検出

フレーム同期の検出を容易にするため、SPCR[1,2]でRINTM = XINTM = 10bに設定してフレーム同期を検出した時に、受信CPU割り込みと送信CPU割り込み(RINTおよびXINT)を発生するようプログラムすることができます。このモードは他の種類のシリアル・ポート割り込みと異なり、シリアル・ポートの対応する部分がリセット状態のときにも動作できます(受信部がリセット状態のときにRINTをアクティブにするなど)。この場合でも、FS(R/X)MとFS(R/X)Pはフレーム同期の適切なソースと極性を選択します。したがって、シリアル・ポートがリセット状態のときでもこれらの信号はCPUクロックに同期され、シリアル・ポートの受信部分と送信部分に入力された時点で、RINTおよびXINTとしてCPUに送られます。その結果、新しいフレーム同期パルスを検出することができ、この後、CPUはシリアル・ポートのリセットを安全に解除できます。

## 2.5.4 クロック・タイミングの例

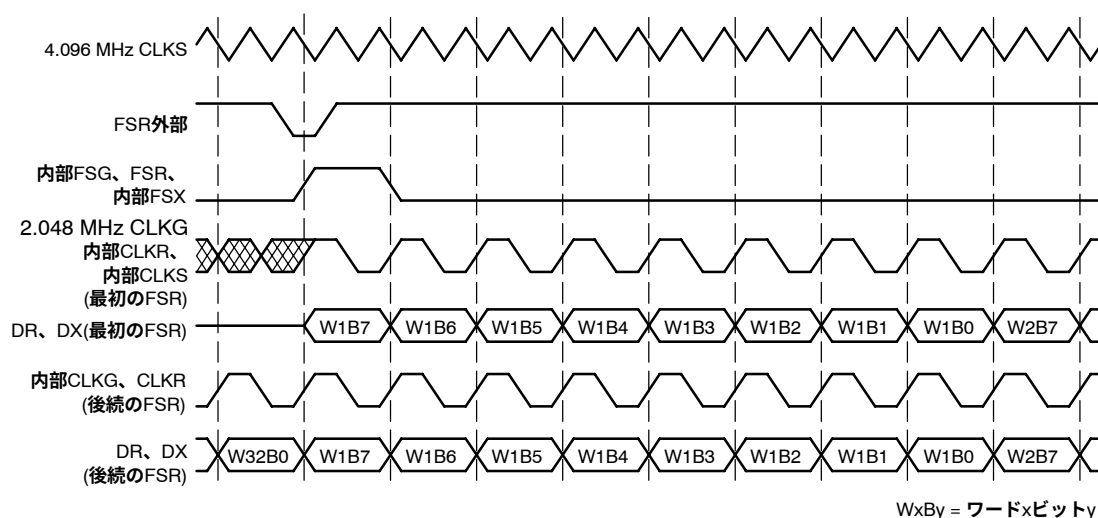
以下の各節に、クロック・タイミングの例を示します。

### 2.5.4.1 ダブルレートSTバス・クロック

図2-48に、Mitel STバス互換として設定されたMcBSPの構成を示します。この場合、最大フレーム周波数で動作することに注意してください。

- ☐ CLK(R/X)M = 1、内部CLK(R/X)がサンプル・レート生成器によって内部で生成されます。
- ☐ GSYNC = 1、CLKGをFSR上の外部フレーム同期信号入力と同期させます。CLKGはフレーム同期信号がアクティブになるまで同期されません。また、パルス幅を最小にするために、FSRは内部で再生成されます。
- ☐ CLKSM = 1、外部クロック(CLKS)がサンプル・レート生成器を駆動します。
- ☐ CLKSP = 1、CLKSの立ち下がりエッジにより、CLKGと内部CLK(R/X)が生成されます。
- ☐ CLKGDV = 1、受信クロック(CLKRとして表示)はCLKS周波数の1/2
- ☐ FS(R/X)P = 1、アクティブ・ローのフレーム同期パルス
- ☐ (R/X)FRLN1 = 11111b、32ワード/フレーム
- ☐ (R/X)WDLEN1 = 0、8ビット・ワード
- ☐ (R/X)PHASE = 0、シングル・フェーズ・フレーム、したがって(R/X)FRLN2 = (R/X)WDLEN2 = X
- ☐ (R/X)DATDLY = 0、データ遅延なし

図2-48. STバスとMVIPの例



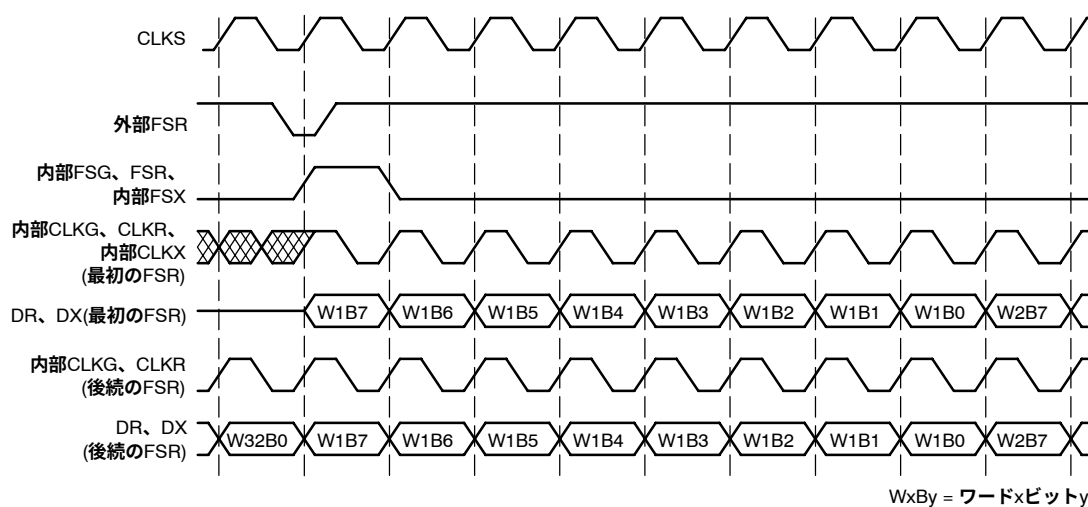
## 2.5.4.2 シングルレートSTバス・クロック

2

この例は、次の点を除いてSTバスの例と同じです。

- ☐ CLKGDV = 0、CLKSが分周されることなく内部CLK(R/X)を駆動します(シングルレート・クロック)。
- ☐ CLKSP = 0、CLKSの立ち上がりエッジにより内部クロックCLKGおよび内部CLK(R/X)が生成されます。

図2-49. シングルレート・クロックの例



外部FSRの検出にはCLKSの立ち上がりエッジが使用されます。この外部フレーム同期パルスは、内部McBSPクロックの再同期と、内部用のフレーム同期の生成に使用されます。内部フレーム同期の生成により、その幅は内部クロックの立ち下がりエッジで十分検出できるようになっています。



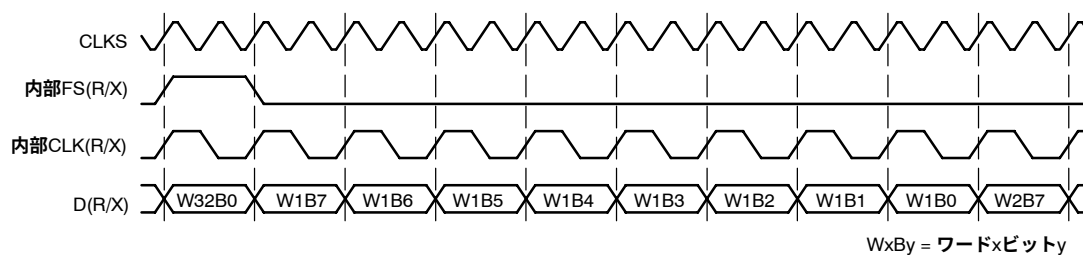
### 2.5.4.3 ダブルレート・クロックの例

この例は、次の点を除いてSTバスの例と同じです。

- ☐ CLKSP = 0、CLKSの立ち上がりエッジによりCLKGとCLK(R/X)が生成されます。
- ☐ CLKGDV = 1、CLKG、内部CLKR、内部CLKXの周波数は、CLKSの1/2です。
- ☐ GSYNC = 0、CLKSがCLKGを生成します。CLKGはフリー・ラン状態で、FSRによって再同期されません。
- ☐ FS(R/X)M = 0、フレーム同期信号は外部から供給されます。フレーム同期パルスは検出に十分な幅を持っています。
- ☐ FS(R/X)P = 0、アクティブ・ハイの入力フレーム同期信号
- ☐ (R/X)DATDLY = 1、1ビットのデータ遅延

2

図2-50. ダブルレート・クロックの例



## 2.6 マルチチャネル選択動作

シングル・フェーズ・フレームでMcBSPを設定することで、送信部および受信部について個別に複数のチャネルを選択できます。各フレームは、時分割多重(TDM)データ・ストリームを表します。(R/X)FRLLEN1で表されるフレーム当たりのワード数は、選択可能なチャネル数を示します。

TDMデータ・ストリームを使用すると、CPUがそのごく一部を処理しなければならない場合があります。したがって、マルチチャネル選択では、メモリーとバスの専有時間を節約するために、送信および受信について個別に特定のチャネルをイネーブルにすることができます。最大128チャネルのビット・ストリーム中、最大32チャネルをイネーブルにすることができます。

受信チャネルがイネーブルではない場合:

- ☐ ワードの最終ビット受信時にRRDYが1にセットされません。
- ☐ ワードの最終ビット受信時にRBR[1,2]がDRR[1,2]にコピーされません。したがって、RRDYはアクティブになりません。これは、このワードについては割り込みまたは同期イベントが生成されないことを意味します。

送信チャネルがイネーブルではない場合:

- ☐ DXはハイ・インピーダンス状態になります。
- ☐ 対応するワードのシリアル送信終了時にDXR[1,2]からXSR[1,2]へのコピーが自動的に行われません。
- ☐ 同様に、 $\overline{\text{XEMPTY}}$ およびXRDYも対応するシリアル・ワードの送信終了によって影響を受けません。

イネーブルにした送信チャネルでは、データをマスクまたは送信することができます。マスクした場合は、送信チャネルがイネーブルであっても、DXピンは強制的にハイ・インピーダンス状態になります。

### 2.6.1 マルチチャネル動作の制御レジスタ

マルチチャネル動作では次の制御レジスタが使用されます。

- 1) マルチチャネル制御1および2(MCR[1,2])レジスタ
- 2) 送信チャネル・イネーブル・パーティションA/B(XCER[A/B])レジスタ
- 3) 受信チャネル・イネーブル・パーティションA/B(RCER[A/B])レジスタ

以下の各節では、マルチチャネル動作の制御におけるこれらのレジスタの使用方法について説明します。

図2-51. マルチチャネル制御レジスタ1(MCR1)

| 15   | 9 | 8 | 7 | 6      | 5      | 4     | 2 | 1    | 0     |
|------|---|---|---|--------|--------|-------|---|------|-------|
| 予約   |   |   |   | RPBBLK | RPABLK | RCBLK |   | 予約   | RMCM  |
| R,+0 |   |   |   | RW,+0  | RW,+0  | R,+0  |   | R,+0 | RW,+0 |

注：R = リード、W = ライト、+0 = リセット時の値

表2-23. マルチチャネル制御レジスタ1(MCR1)のビット/フィールドの説明

| ビット  | 名前     | 機能                                       | 節     |
|------|--------|------------------------------------------|-------|
| 15-9 | 予約     | 予約                                       |       |
| 8-7  | RPBBLK | 受信パーティションBブロック                           | 2.6.3 |
|      |        | RPBBLK = 00      ブロック1。チャンネル16～チャンネル31   |       |
|      |        | RPBBLK = 01      ブロック3。チャンネル48～チャンネル63   |       |
|      |        | RPBBLK = 10      ブロック5。チャンネル80～チャンネル95   |       |
|      |        | RPBBLK = 11      ブロック7。チャンネル112～チャンネル127 |       |
| 6-5  | RPABLK | 受信パーティションAブロック                           | 2.6.3 |
|      |        | RPABLK = 00      ブロック0。チャンネル0～チャンネル15    |       |
|      |        | RPABLK = 01      ブロック2。チャンネル32～チャンネル47   |       |
|      |        | RPABLK = 10      ブロック4。チャンネル64～チャンネル79   |       |
|      |        | RPABLK = 11      ブロック6。チャンネル96～チャンネル111  |       |

## マルチチャネル選択動作

表2-23. マルチチャネル制御レジスタ1(MCR1)のビット/フィールドの説明(続き)

| ビット | 名前    | 機能                                                                                  | 節       |
|-----|-------|-------------------------------------------------------------------------------------|---------|
| 4-2 | RCBLK | 受信カレント・ブロック                                                                         | 2.6.3.2 |
|     |       | RCBLK = 000      ブロック0。チャンネル0～チャンネル15                                               |         |
|     |       | RCBLK = 001      ブロック1。チャンネル16～チャンネル31                                              |         |
|     |       | RCBLK = 010      ブロック2。チャンネル32～チャンネル47                                              |         |
|     |       | RCBLK = 011      ブロック3。チャンネル48～チャンネル63                                              |         |
|     |       | RCBLK = 100      ブロック4。チャンネル64～チャンネル79                                              |         |
|     |       | RCBLK = 101      ブロック5。チャンネル80～チャンネル95                                              |         |
|     |       | RCBLK = 110      ブロック6。チャンネル96～チャンネル111                                             |         |
|     |       | RCBLK = 111      ブロック7。チャンネル112～チャンネル127                                            |         |
| 1   | 予約    | 予約                                                                                  |         |
| 0   | RMCM  | 受信マルチチャネル選択イネーブル                                                                    | 2.6.2   |
|     |       | RMCM = 0      全128チャンネルがイネーブル                                                       |         |
|     |       | RMCM = 1      全チャンネルがデフォルトでディセーブル。必要なチャンネルは、RP(A/B)BLKおよびRCER(A/B)を適宜イネーブルにして選択します。 |         |

図2-52. マルチチャネル制御レジスタ2(MCR2)

|      |    |    |    |    |    |   |        |   |        |   |       |   |       |   |   |
|------|----|----|----|----|----|---|--------|---|--------|---|-------|---|-------|---|---|
| 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8      | 7 | 6      | 5 | 4     | 3 | 2     | 1 | 0 |
| 予約   |    |    |    |    |    |   | XPBBLK |   | XPABLK |   | XCBLK |   | XMCM  |   |   |
| R,+0 |    |    |    |    |    |   | RW,+0  |   | RW,+0  |   | R,+0  |   | RW,+0 |   |   |

注: R = リード、W = ライト、+0 = リセット時の値

表2-24. マルチチャネル制御レジスタ2(MCR2)のビット/フィールドの説明

| ビット  | 名前     | 機能                                       | 節       |
|------|--------|------------------------------------------|---------|
| 15-9 | 予約     | 予約                                       |         |
| 8-7  | XPABLK | 送信パーティションAブロック                           | 2.6.3   |
|      |        | XPABLK = 00      ブロック0。チャンネル0～チャンネル15    |         |
|      |        | XPABLK = 01      ブロック2。チャンネル32～チャンネル47   |         |
|      |        | XPABLK = 10      ブロック4。チャンネル64～チャンネル79   |         |
|      |        | XPABLK = 11      ブロック6。チャンネル96～チャンネル111  |         |
| 6-5  | XPBBLK | 送信パーティションBブロック                           | 2.6.3   |
|      |        | XPBBLK = 00      ブロック1。チャンネル16～チャンネル31   |         |
|      |        | XPBBLK = 01      ブロック3。チャンネル48～チャンネル63   |         |
|      |        | XPBBLK = 10      ブロック5。チャンネル80～チャンネル95   |         |
|      |        | XPBBLK = 11      ブロック7。チャンネル112～チャンネル127 |         |
| 4-2  | XCBLK  | 送信カレント・ブロック                              | 2.6.3.2 |
|      |        | XCBLK = 000      ブロック0。チャンネル0～チャンネル15    |         |
|      |        | XCBLK = 001      ブロック1。チャンネル16～チャンネル31   |         |
|      |        | XCBLK = 010      ブロック2。チャンネル32～チャンネル47   |         |
|      |        | XCBLK = 011      ブロック3。チャンネル48～チャンネル63   |         |
|      |        | XCBLK = 100      ブロック4。チャンネル64～チャンネル79   |         |
|      |        | XCBLK = 101      ブロック5。チャンネル80～チャンネル95   |         |
|      |        | XCBLK = 110      ブロック6。チャンネル96～チャンネル111  |         |
|      |        | XCBLK = 111      ブロック7。チャンネル112～チャンネル127 |         |

表2-24. マルチチャネル制御レジスタ2(MCR2)のビット/フィールドの説明 (続き)

| ビット | 名前   | 機能                                                                                                                                                                | 節     |
|-----|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 1-0 | XMCM | 送信マルチチャネル選択イネーブル                                                                                                                                                  | 2.6.2 |
|     |      | XMCM = 00 全チャンネルがマスクなしでイネーブル(データ送信中はDXピンが常にドライブされます <sup>†</sup> )。                                                                                               |       |
|     |      | XMCM = 01 全チャンネルがディセーブルであり、デフォルトでマスクされます。必要なチャンネルは、XP(A/B)BLKおよびXCER(A/B)で適宜イネーブルにして選択します。また、選択したチャンネルはマスクされないため、DXピンは常にドライブされます。                                 |       |
|     |      | XMCM = 10 全チャンネルがイネーブルですが、マスクされます。XP(A/B)BLKおよびXCER(A/B)によって選択したチャンネルはマスク解除されます。                                                                                  |       |
|     |      | XMCM = 11 全チャンネルがディセーブルであり、デフォルトでマスクされます。必要なチャンネルは、RP(A/B)BLKおよびRCER(A/B)を適宜イネーブルにして選択します。選択したチャンネルは、RP(A/B)BLKおよびXCER(A/B)によってマスク解除できます。このモードは、対称型の送受信動作に使用されます。 |       |

<sup>†</sup> DXは、(a)パケット間インターバル、(b)チャンネルがイネーブルかどうかにかかわらず、マスクされている場合、(c)チャンネルがディセーブルの場合においては、マスクされるか、ハイ・インピーダンスになります。

### 2.6.2 マルチチャネル選択のイネーブル

マルチチャネル・モードは、受信および送信について、MCR[1,2]でそれぞれRMCM = 1およびXMCMを0以外の値に設定することで、個別にイネーブルにできます。

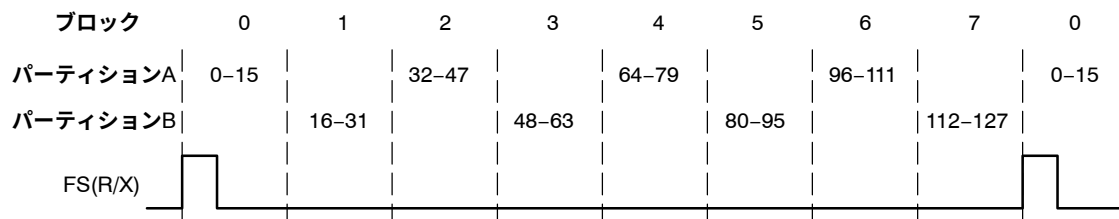
### 2.6.3 チャンネルのマスクとイネーブル

使用可能な128チャンネルのうち、合計32チャンネルまでを任意のタイミングでイネーブルにすることができます。128のチャンネルは8つのブロック(0~7)で構成されており、各ブロックは16の連続チャンネルをもっています。さらに、偶数番号のブロック0、2、4、および6はパーティションAに属し、奇数番号のブロック1、3、5、7はパーティションBに属します。

イネーブルにするチャンネルの数をフレームの途中で更新して、チャンネルの任意のグループをイネーブルにすることができます。この更新は、それぞれ16の連続チャンネルから成る2つのブロック(1つの奇数ブロックと1つの偶数ブロック)を交互に使用するために、フレーム内の任意の時点で行うことができます。一方のブロックはパーティションAに、他方のブロックはパーティションBに属します。

8つの16チャネル・ブロックから任意の2つを選び、最大で合計32チャネルをイネーブルにすることができます。図2-53に示すように、フレーム内の16チャネルごとにブロックが割り当てられます。MCR[1,2]の(R/X)PABLKフィールドと(R/X)PBBLKフィールドが、それぞれパーティションAおよびBで選択するブロックを決定します。このイネーブルの設定は、送信および受信について個別に行うことができます。

図2-53. パーティションAおよびBにおけるブロックによるチャネルのイネーブル



送信データをマスクすることで、送信についてイネーブルにしたチャネルでの送信期間中に、そのDXピンをハイ・インピーダンス状態にすることができます。ソフトウェア上、対称型の送受信を用いた方が都合がいい場合は、この機能によって共有シリアル・バスで送信チャネルをディセーブルにすることができます。受信については、複数の受信がシリアル・バスの競合を引き起こすことはないため、同様の機能は必要ありません。

注：

DXは、(a)パケット間インターバル、(b)チャネルがイネーブルかどうかにかかわらず、マスクされている場合、(c)チャネルがディセーブルの場合においては、マスクされるか、ハイ・インピーダンスになります。

次に、各XMCM値での送信中のマルチチャネル動作について説明します。

- ☐ XMCM = 00b: シリアル・ポートは、XFRLEN1でプログラムされたワード数のデータを、DXピン経由で送信します。したがって、送信中はDXピンがドライブされます。
- ☐ XMCM = 01b: 希望のチャネル、つまり送信する必要があるワードだけが、XP(A/B)BLKおよびXCER(A/B)によって選択されます。したがって、それらの選択されたワードだけがDXR[1,2]にライトされ、最終的に送信されます。XINTM = 00bの場合、DXR1からXSR1へのコピーが行われるたびにXINTが生成され、そのXINTの数は、XCER(A/B)によって選択されたチャネルの数に等しくなります(XFRLEN1には等しくなりません)。
- ☐ XMCM = 10b: この場合、すべてのチャネルがイネーブルになります。つまり、データ・フレーム(XFRLEN1)の全ワードがDXR[1,2]にライトされ、そのたびにDXR[1,2]からXSR[1,2]へのコピーが行われます。しかし、DXピンはXP(A/B)BLKおよびXCER(A/B)によって選択されたチャネルについてのみドライブされ、その他の場合はハイ・インピーダンスになります。この場合、XINTM = 00bであれば、DXR1からXSR1へのコピーごとに生成される割り込みの数は、そのフレームのワード数(XFRLEN1)に等しくなります。

- XMCM = 11b: このモードは基本的に、XMCM = 01bと10bの場合を組み合わせ、対称型の送受信動作が行われるようにしたものです。受信の場合は、RP(A/B)BLKおよびRCER(A/B)によって選択されたチャネルについてのみRBR[1,2]からDRR[1,2]へのコピーが行われます。RBR[1,2]からDRR[1,2]へのコピーごとにRINTが生成される場合は、RINTの回数はRCER(A/B)によって選択されたチャネルの数に等しくなります(RFRLN1でプログラムされたワード数には等しくなりません)。送信の場合は、対称を保つために受信と同じブロックが使用されるため、XP(A/B)BLKは参照されません。RP(A/B)BLKで選択されたすべてのチャネルについて、DXR[1,2]にデータがロードされ、DXR[1,2]からXSR[1,2]へのコピーが行われます。しかし、DXピンはXCER(A/B)によって選択されたチャネルについてのみドライブされます。XCER(A/B)でイネーブルにするチャネルは、RCER(A/B)で選択するチャネルと同じか、そのチャネルの一部に限られます。したがって、XINTM = 00bの場合、CPUへの送信割り込みは(XCER[A/B]ではなく)RCER(A/B)で選択されたチャネルと同じ数だけ生成されます。

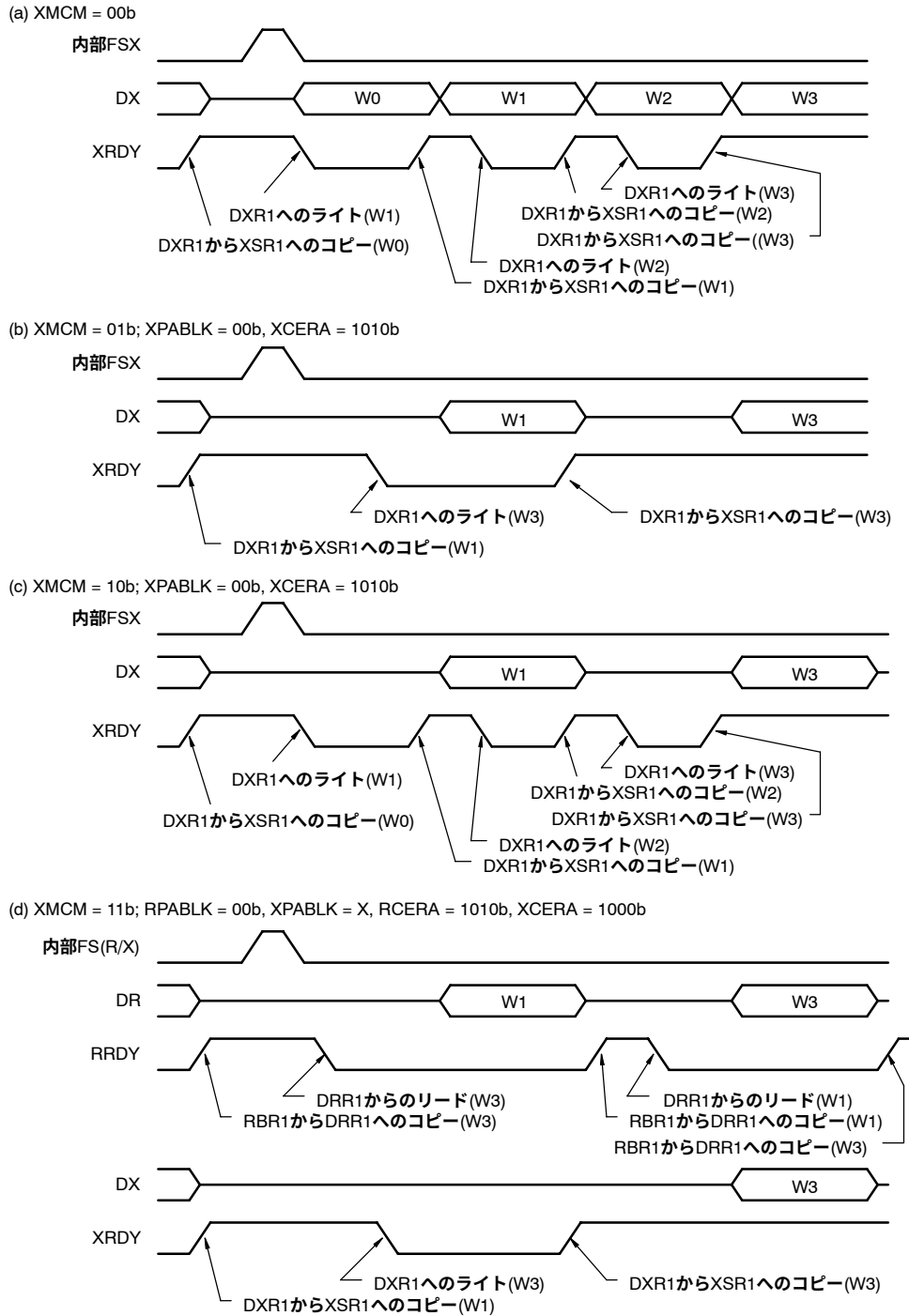
図2-54に、これまで説明したすべてのモードにおける、次の条件でのMcBSPピン上の動作状況を示します。

- (R/X)PHASE = 0、マルチチャネル選択をイネーブルにするためシングル・フェーズ・フレームを選択
- FRLEN1 = 011b、4ワード・フレーム
- WDLN1 = 000b、8ビット・ワード

以下の各図では、各種イベントの発生箇所を示す矢印は例示にすぎません。可能な限り、それらのイベントが発生できるタイミングを表示します。



図2-54. XMCMの動作



## 2.6.3.1 チャネル・イネーブル・レジスタ: (R/X)CER(A/B)

受信チャネル・イネーブル・パーティションAおよびBレジスタ(RCER[A/B])と送信チャネル・イネーブル・パーティションAおよびBレジスタ(XCER[A/B])は、それぞれ受信および送信について、32チャネルのうち任意のチャネルをイネーブルにするために使用されます。32チャネルのうち、16チャネルはパーティションAのブロックに、残りの16チャネルはパーティションBのブロックに属します。これらを、図2-55、図2-56、図2-57、および図2-58に示します。

表2-25、表2-26、表2-27、および表2-28に示す(R/X)CERAおよび(R/X)CERBレジスタ・フィールドは、それぞれパーティションAおよびBの16チャネル幅のブロック内の各チャネルをイネーブルにします。MCRの(R/X)PABLKフィールドと(R/X)PBBLKフィールドは、イネーブルにする16チャネル・ブロックを選択します。

図2-55. 受信チャネル・イネーブル・レジスタ・パーティションA(RCERA)

|        |        |        |        |        |        |       |       |
|--------|--------|--------|--------|--------|--------|-------|-------|
| 15     | 14     | 13     | 12     | 11     | 10     | 9     | 8     |
| RCEA15 | RCEA14 | RCEA13 | RCEA12 | RCEA11 | RCEA10 | RCEA9 | RCEA8 |
| RW,+0  | RW,+0  | RW,+0  | RW,+0  | RW,+0  | RW,+0  | RW,+0 | RW,+0 |
| 7      | 6      | 5      | 4      | 3      | 2      | 1     | 0     |
| RCEA7  | RCEA6  | RCEA5  | RCEA4  | RCEA3  | RCEA2  | RCEA1 | RCEA0 |
| RW,+0  | RW,+0  | RW,+0  | RW,+0  | RW,+0  | RW,+0  | RW,+0 | RW,+0 |

注: R = リード、W = ライト、+0 = リセット時の値

表2-25. 受信チャネル・イネーブル・レジスタ・パーティションA(RCERA)のビット/フィールドの説明

| ビット  | 名前           | 機能                                                  |
|------|--------------|-----------------------------------------------------|
| 15-0 | RCEA(0:15)   | 受信チャネル・イネーブル                                        |
|      | RCEA $n$ = 0 | パーティションAに属する偶数番号ブロックにおける $n$ 番目のチャネルの受信をディセーブルにします。 |
|      | RCEA $n$ = 1 | パーティションAに属する偶数番号ブロックにおける $n$ 番目のチャネルの受信をイネーブルにします。  |

図2-56. 受信チャネル・イネーブル・レジスタ・パーティションB(RCERB)

|        |        |        |        |        |        |       |       |
|--------|--------|--------|--------|--------|--------|-------|-------|
| 15     | 14     | 13     | 12     | 11     | 10     | 9     | 8     |
| RCEB15 | RCEB14 | RCEB13 | RCEB12 | RCEB11 | RCEB10 | RCEB9 | RCEB8 |
| RW,+0  | RW,+0  | RW,+0  | RW,+0  | RW,+0  | RW,+0  | RW,+0 | RW,+0 |
| 7      | 6      | 5      | 4      | 3      | 2      | 1     | 0     |
| RCEB7  | RCEB6  | RCEB5  | RCEB4  | RCEB3  | RCEB2  | RCEB1 | RCEB0 |
| RW,+0  | RW,+0  | RW,+0  | RW,+0  | RW,+0  | RW,+0  | RW,+0 | RW,+0 |

注: R = リード、W = ライト、+0 = リセット時の値

表2-26. 受信チャネル・イネーブル・レジスタ・パーティションB(RCERB)のビット/フィールドの説明

| ビット  | 名前           | 機能                                                   |
|------|--------------|------------------------------------------------------|
| 15-0 | RCEB(0:15)   | 受信チャネル・イネーブル                                         |
|      | RCEB $n$ = 0 | パーティションBに属する奇数番号ブロックにおける $n$ 番目のチャンネルの受信をディセーブルにします。 |
|      | RCEB $n$ = 1 | パーティションBに属する奇数番号ブロックにおける $n$ 番目のチャンネルの受信をイネーブルにします。  |

図2-57. 送信チャネル・イネーブル・レジスタ・パーティションA(XCERA)

|        |        |        |        |        |        |       |       |
|--------|--------|--------|--------|--------|--------|-------|-------|
| 15     | 14     | 13     | 12     | 11     | 10     | 9     | 8     |
| XCEA15 | XCEA14 | XCEA13 | XCEA12 | XCEA11 | XCEA10 | XCEA9 | XCEA8 |
| RW,+0  | RW,+0  | RW,+0  | RW,+0  | RW,+0  | RW,+0  | RW,+0 | RW,+0 |
| 7      | 6      | 5      | 4      | 3      | 2      | 1     | 0     |
| XCEA7  | XCEA6  | XCEA5  | XCEA4  | XCEA3  | XCEA2  | XCEA1 | XCEA0 |
| RW,+0  | RW,+0  | RW,+0  | RW,+0  | RW,+0  | RW,+0  | RW,+0 | RW,+0 |

注: R = リード、W = ライト、+0 = リセット時の値

表2-27. 送信チャネル・イネーブル・レジスタ・パーティションA(XCERA)のビット/フィールドの説明

| ビット  | 名前           | 機能                                                   |
|------|--------------|------------------------------------------------------|
| 15-0 | XCEA(0:15)   | 送信チャネル・イネーブル                                         |
|      | XCEA $n$ = 0 | パーティションAに属する偶数番号ブロックにおける $n$ 番目のチャンネルの送信をディセーブルにします。 |
|      | XCEA $n$ = 1 | パーティションAに属する偶数番号ブロックにおける $n$ 番目のチャンネルの送信をイネーブルにします。  |

## マルチチャネル選択動作

図2-58. 送信チャネル・イネーブル・レジスタ・パーティションB(XCERB)

|        |        |        |        |        |        |       |       |
|--------|--------|--------|--------|--------|--------|-------|-------|
| 15     | 14     | 13     | 12     | 11     | 10     | 9     | 8     |
| XCEB15 | XCEB14 | XCEB13 | XCEB12 | XCEB11 | XCEB10 | XCEB9 | XCEB8 |
| RW,+0  | RW,+0  | RW,+0  | RW,+0  | RW,+0  | RW,+0  | RW,+0 | RW,+0 |
| 7      | 6      | 5      | 4      | 3      | 2      | 1     | 0     |
| XCEB7  | XCEB6  | XCEB5  | XCEB4  | XCEB3  | XCEB2  | XCEB1 | XCEB0 |
| RW,+0  | RW,+0  | RW,+0  | RW,+0  | RW,+0  | RW,+0  | RW,+0 | RW,+0 |

注: R = リード、W = ライト、+0 = リセット時の値

表2-28. 送信チャネル・イネーブル・レジスタ・パーティションB(XCERB)のビット/フィールドの説明

| ビット  | 名前           | 機能                                                  |
|------|--------------|-----------------------------------------------------|
| 15-0 | XCEB(0:15)   | 送信チャネル・イネーブル                                        |
|      | XCEB $n$ = 0 | パーティションBに属する奇数番号ブロックにおける $n$ 番目のチャネルの送信をディセーブルにします。 |
|      | XCEB $n$ = 1 | パーティションBに属する奇数番号ブロックにおける $n$ 番目のチャネルの送信をイネーブルにします。  |

### 2.6.3.2 チャネル選択の変更

マルチチャネル選択機能を使って、32チャネルのある決まったグループをイネーブルにすることができます。このグループは、チャネルの割り当ての変更が必要になるまで、CPUの介入なしでイネーブルのままにしておくことができます。ブロック終了の割り込みに応答してフレームの途中でブロック割り当てレジスタを更新すれば、フレーム内の任意の数、グループ、またはすべてのワード/チャネルにアクセスできます(第2.6.3.3節「割り込みの更新」参照)。

注:

チャネル選択を変更するときは、現在選択されているブロックに影響を及ぼさないよう、ユーザーの方で注意する必要があります。

現在選択されているブロックは、受信の場合はMCR1のRCBLKフィールド、送信の場合はMCR2のXCBLKフィールドを通じてリードすることができます。(R/X)P(A/B)BLKレジスタの選択しているブロックが現在のブロックを指している場合は、対応するチャネル・イネーブル・レジスタは変更できません。同様に、MCR[1,2]の(R/X)PABLKフィールドと(R/X)PBBLKフィールドも、現在選択されているブロックを指している場合、あるいはそれらを指すよう変更しようとしている場合は、変更できません。チャネルの総数が16以下の場合は、現在のパーティションが常に指し示されます。この場合、シリアル・ポートのリセットによってのみイネーブルを変更できます。

### 2.6.3.3 割り込みの更新

マルチチャネル動作中には、SPCR[1,2]でRINTM = 01bまたはXINTM = 01bであれば、16チャネルのブロック境界ごとにCPUへの受信割り込み(RINT)または送信割り込み(XINT)がそれぞれ発生します。この割り込みは、新しいパーティションに移ったことを示します。この場合は、現在のパーティションをチェックして、パーティションAまたはBのブロック選択が現在のブロックを指していなければ、ブロック選択を変更することができます。これらの割り込みは、2 CPUクロック幅のアクティブ・ハイ・パルスです。(R/X)MCM = 0(非マルチチャネル動作)のときは、RINTM = XINTM = 01bであっても割り込みが発生しません。

### 2.6.4 A-bisインターフェイス機能('C5410のみで使用可能)

A-bisモード(ABIS = 1)では、McBSPはPCMリンク上で最大1024ビットを送受信できます。受信部分は設定された受信パターンに従って1024ビットのPCMフレームから1024ビットすべてを抽出し、DRRが有効な16個のビット・ワードで満たされたとき、または受信フレームが終了したときに、CPUへの割り込みを生成することができます。また、送信部分は設定された送信パターンに従って特定の位置で1024ビットのPCMフレームに最大1024ビットを伸張し、16ビット・ワードが送信されたか、送信フレームが終了したときに割り込みを生成することができます。このモードでは、ワード長を16ビット・モードに設定する必要があります(WDLEN1 = 010b)。そうしないと、A-bisモードにはなりません。

A-bisモードでは、チャネル・イネーブル・レジスタ(R/X)CER(A/B)は通常のマルチチャネル動作とは異なる機能を持ちます。A-bisモードでは、これらのレジスタはどのチャネルをイネーブルにするかを示すのではなく、データ・ストリームのどのビットをイネーブルにするかを示します。(R/X)CER(A/B)レジスタの希望の位置に1をセットすると、受信または送信データ・ストリームにおいて対応するビットがイネーブルになります。

受信部では、RCER(A/B)レジスタによってイネーブルになっていないビットは無視され、受信部で受信されません。イネーブルにされたビットは受信され、有効なデータ・ワードに圧縮されます。有効な16個のデータ・ビットを受信すると、受信したワードがRSR1からDRR1にコピーされ、McBSPがCPUへの割り込みを生成します。RCERAとRCERBは、16サイクルの受信クロックのそれぞれについて、受信マスク・パターンを交互に指定します。図2-59に、受信部のビット・シーケンスの例を示します。

図2-59. A-bisモードの受信動作

‡ DRR1の受信データ

† DXピンから送信されるデータ

2-84 マルチチャネル・バッファド・シリアル・ポート (McBSP)

動作中、2つのDMAチャネル(1つは送信用、1つは受信用)が(R/X)CER(A/B)へのパターン選択データの更新に使用されます。メモリー内の1つの16ワード・ブロックに、受信部のビット・パターン選択がライトされています。16ビットずつの16のワードには、256ビットPCMリンクの受信選択パターン全体がライトされています。REVTAイベントが発生するたびに、DMAは必要に応じて新しい受信選択パターン・データをメモリーからRCER(A/B)にコピーし、その受信選択パターン参照のレジスタをRCERAからRCERBへ、またはその逆に自動的に切り替えます。言い換えれば、DMAチャネルの転送先は当初、RCERAに設定されています。RCERAに最初にアクセスした後、転送先が自動的にRCERBに切り替わります。次にRCERBにアクセスした後、転送先が自動的にRCERAに戻ります。RCERAとRCERBの切り替えは自動的に処理されるため、転送先アドレスを変更するようDMAコントローラを設定する必要はありません。A-bis転送中は、受信部はRCERAとRCERBの使用を切り替えて、16シリアル・ポート・クロックの各グループのイネーブル・パターンを指定します。

メモリー内の別の16ワード・ブロックには、送信部のビット・パターン選択がライトされています。16ビットずつの16のワードには、256ビットPCMリンクの送信選択パターン全体がライトされています。XEVTAイベントが発生するたびに、DMAは必要に応じて新しい送信選択パターン・データをメモリーからXCER(A/B)にコピーし、その送信選択パターン参照のレジスタをXCERAからXCERBへ、またはその逆に自動的に切り替えます。言い換えれば、DMAチャネルの転送先は当初、XCERAに設定されています。XCERAに最初にアクセスした後、転送先が自動的にXCERBに切り替わります。次にXCERBにアクセスした後、転送先が自動的にXCERAに戻ります。XCERAとXCERBの切り替えは自動的に処理されるため、転送先アドレスを変更するようDMAコントローラを設定する必要はありません。A-bis転送中、送信部はXCERAとXCERBの使用を切り替えて、16シリアル・ポート・クロックの各グループのイネーブル・パターンを指定します。

結果、送信部と受信部の両方に一意の選択パターンを用いた、連続した256ビットのデータ送受信が実現されます。

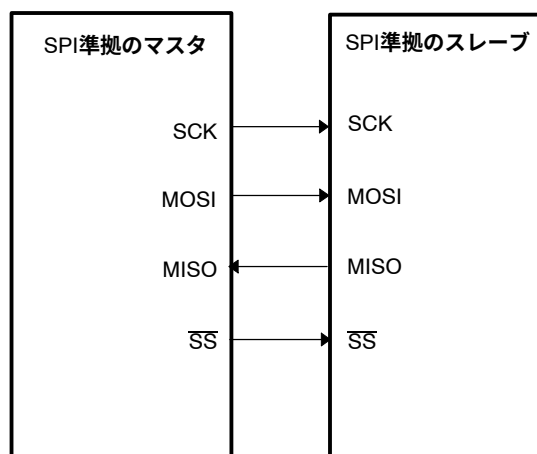
## 2.7 SPIプロトコル: McBSPクロック・ストップ・モード

SPIプロトコルは、1つのマスタ・デバイスと1つまたは複数のスレーブ・デバイスによるマスタ・スレーブ構成になります。インターフェイスは次の4つの信号で構成されます。

- ☐ シリアル・データ入力(マスタ・イン・スレーブ・アウト、MISO)
- ☐ シリアル・データ出力(マスタ・アウト・スレーブ・イン、MOSI)
- ☐ シフト・クロック(SCK)
- ☐ スレーブ・イネーブル信号( $\overline{SS}$ )

図2-61に、1つのスレーブ・デバイスを持つ一般的なSPIインターフェイスを示します。

図2-61. 一般的なSPIインターフェイス



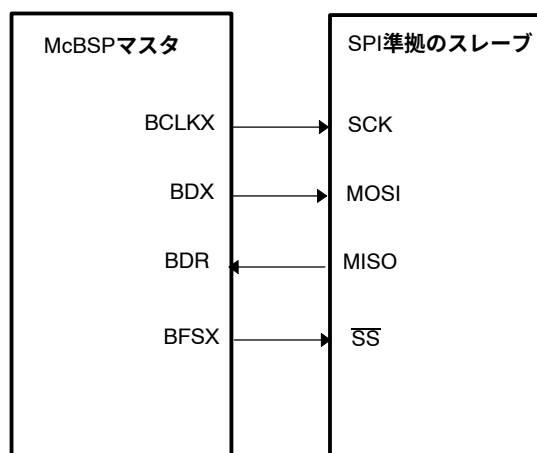
マスタ・デバイスは、シフト・クロックとスレーブ・イネーブル信号を出力して、通信の流れを制御します。スレーブ・イネーブル信号は、スレーブ・デバイス(クロックを出力しないデバイス)のシリアル・データ入出力をイネーブルにする、任意のアクティブ・ロー信号です。専用のスレーブ・イネーブル信号がない場合は、マスタとスレーブの間の通信はシフト・クロックの有無によって決定されます。そのような構成では、スレーブ・デバイスが常にイネーブルである必要があり、複数のスレーブは使用できません。

McBSPのクロック・ストップ・モードは、SPIプロトコルとの互換性を備えています。McBSPをクロック・ストップ・モードに設定すると、McBSPがSPIマスタ・デバイスまたはスレーブ・デバイスとして動作するよう、送信部と受信部が内部で同期します。送信クロック信号(BCLKX)はSPIプロトコルのシリアル・クロック信号(SCK)に対応し、送信フレーム同期信号(BFSX)はスレーブ・イネーブル信号( $\overline{SS}$ )として使用されます。クロック・ストップ・モードでは、受信クロック信号(BCLKR)と受信フレーム同期信号(BFSR)は、送信側で対応するBCLKXとBFSXに内部で接続されるため使用されません。



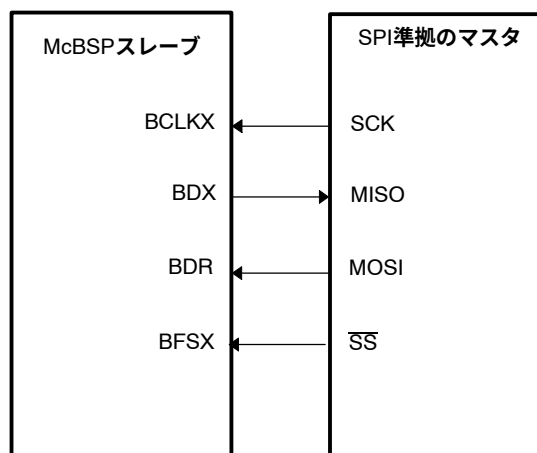
McBSPをマスタとして設定すると、送信出力信号(BDX)がSPIプロトコルのMOSI信号として使用され、受信入力信号(BDR)がMISO信号として使用されます。図2-62に、McBSPをマスタとして使用するSPIインターフェイスを示します。

図2-62. SPI構成: マスタとしてのMcBSP



同様に、McBSPをスレーブとして設定すると、BDXがMISO信号として使用され、BDRがMOSI信号として使用されます。図2-63に、McBSPをスレーブとして使用するSPIインターフェイスを示します。

図2-63. SPI構成: スレーブとしてのMcBSP



## 2.7.1 クロック・ストップ・モードの設定と信号の説明

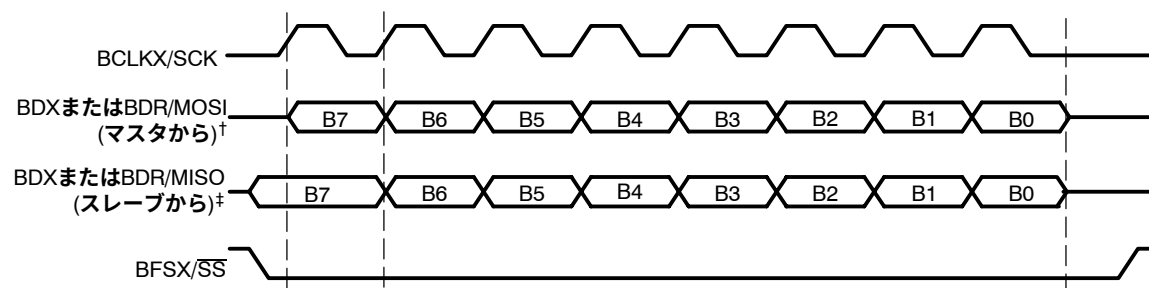
クロック・ストップ・モードの設定には、シリアル・ポート制御レジスタ1(SPCR1)のCLKSTPビット/フィールドと、ピン制御レジスタ(PCR)のCLKXPビットを使用します。CLKSTPビット/フィールドはクロック・ストップ・モードをイネーブルにし、使用可能な2つのタイミング・バリエーションのうち1つを選択します。一方CLKXPビットは、BCLKX信号の極性を設定します。これらのビットの組み合わせにより、表2-29に示すように、4つの異なるクロック・ストップ・モードの設定が可能です。

表2-29. クロック・ストップ・モードの設定

| CLKSTP | CLKXP | クロック動作                                                                             |
|--------|-------|------------------------------------------------------------------------------------|
| 0X     | X     | クロック・ストップ・モードがディセーブル。クロックは非SPIモードであり、イネーブル                                         |
| 10     | 0     | 遅延なしでインアクティブ・ロー: McBSPはCLKXの立ち上がりエッジでデータを送信し、CLKRの立ち下がりエッジでデータを受信します。              |
| 11     | 0     | 遅延をともないインアクティブ・ロー: McBSPはCLKXの立ち上がりエッジよりも1/2サイクル早くデータを送信し、CLKRの立ち上がりエッジでデータを受信します。 |
| 10     | 1     | 遅延なしでインアクティブ・ハイ: McBSPはCLKXの立ち下がりエッジでデータを送信し、CLKRの立ち上がりエッジでデータを受信します。              |
| 11     | 1     | 遅延をともないインアクティブ・ハイ: McBSPはCLKXの立ち下がりエッジよりも1/2サイクル早くデータを送信し、CLKRの立ち下がりエッジでデータを受信します。 |

PCRのCLKXMビット/フィールドにより、BCLKX信号を出力モード(マスタ)または入力モード(スレーブ)に設定して、McBSPをマスタまたはスレーブとして指定します。使用可能な4つのクロック・ストップ・モード設定のタイミング図を、図2-64から図2-67に示します。

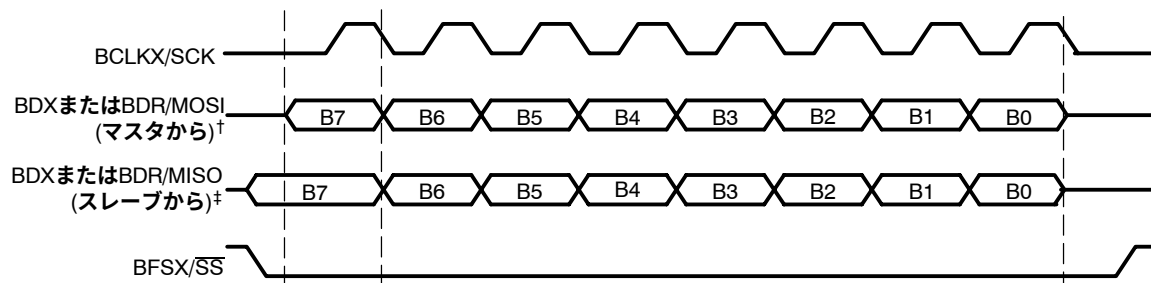
図2-64. CLKSTP = 10bおよびCLKXP = 0でのSPI転送



† McBSPがSPIマスタ(CLKXM = 1)の場合は、MOSI = BDX。McBSPがSPIスレーブ(CLKXM = 0)の場合は、MOSI = BDR。

‡ McBSPがSPIマスタ(CLKXM = 1)の場合は、MISO = BDR。McBSPがSPIスレーブ(CLKXM = 0)の場合は、MISO = BDX。

図2-65. CLKSTP = 11b、CLKXP = 0でのSPI転送

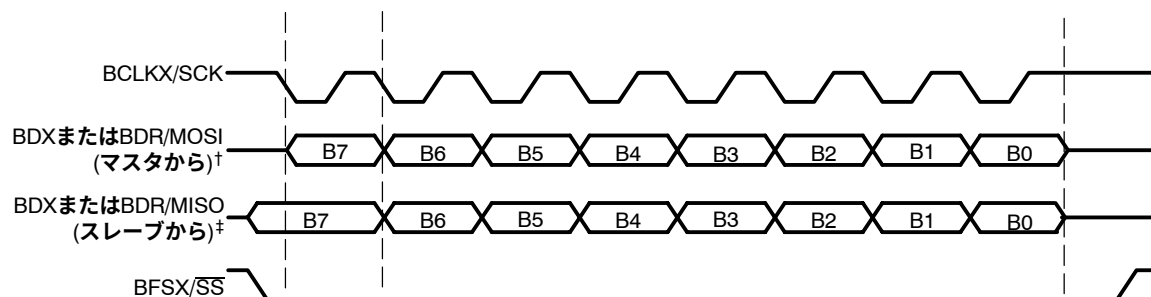


2

† McBSPがSPIマスタ(CLKXM = 1)の場合は、MOSI = BDX。McBSPがSPIスレーブ(CLKXM = 0)の場合は、MOSI = BDR。

‡ McBSPがSPIマスタ(CLKXM = 1)の場合は、MISO = BDR。McBSPがSPIスレーブ(CLKXM = 0)の場合は、MISO = BDX。

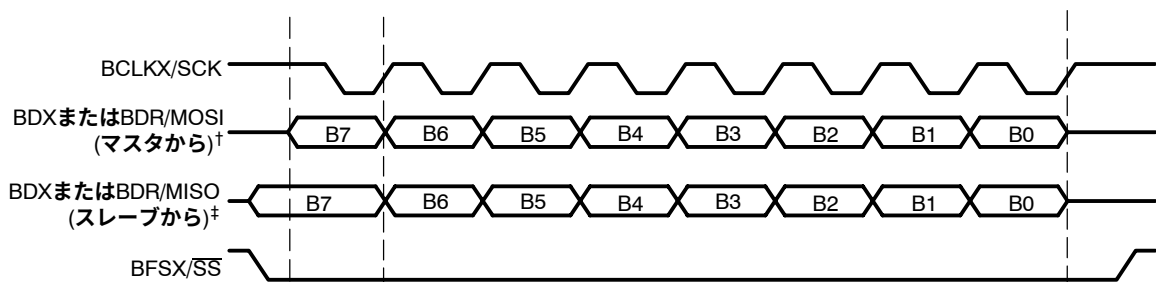
図2-66. CLKSTP = 10b、CLKXP = 1でのSPI転送



† McBSPがSPIマスタ(CLKXM = 1)の場合は、MOSI = BDX。McBSPがSPIスレーブ(CLKXM = 0)の場合は、MOSI = BDR。

‡ McBSPがSPIマスタ(CLKXM = 1)の場合は、MISO = BDR。McBSPがSPIスレーブ(CLKXM = 0)の場合は、MISO = BDX。

図2-67. CLKSTP = 11b、CLKXP = 1でのSPI転送



† McBSPがSPIマスタ(CLKXM = 1)の場合は、MOSI = BDX。McBSPがSPIスレーブ(CLKXM = 0)の場合は、MOSI = BDR。

‡ McBSPがSPIマスタ(CLKXM = 1)の場合は、MISO = BDR。McBSPがSPIスレーブ(CLKXM = 0)の場合は、MISO = BDX。

クロック・ストップ・モードで使用されるフレーム同期信号は、送信全体を通じてアクティブになります。フレーム同期波形の詳細については、次の2つの節で説明します。タイミング図では8ビットの転送を示しましたが、パケット当たり8、12、16、20、24、または32ビットのパケット長を設定できます。パケット長は、受信制御レジスタ1(RCR1)の受信ワード長(RWDLEN1)と、送信制御レジスタ1(XCR1)の送信ワード長(XWDLEN1)によって選択します。クロック・ストップ・モードでは、McBSPの送信回路と受信回路が1つのクロックに同期されるため、RWDLEN1とXWDLEN1を同じ値に設定する必要があります。表2-30に、McBSPをSPIデバイスとして設定するために必要なビット/フィールドを示します。

表2-30. SPIモード設定用のレジスタ・ビット値

| ビット/フィールド | 値        | 説明                                                 | レジスタ  | 参照           |
|-----------|----------|----------------------------------------------------|-------|--------------|
| CLKSTP    | 1xb      | クロック・ストップ・モードをイネーブルにし、2つのタイミング・バリエーションのどちらかを選択します。 | SPCR1 | 2-7ページの表2-4  |
| CLKXP     | 0または1    | BCLKX信号の極性を設定します。                                  | PCR   | 2-12ページの表2-6 |
| CLKXM     | 0または1    | BCLKX信号を入力(スレーブ)または出力(マスタ)として設定します。                | PCR   | 2-12ページの表2-6 |
| RWDLEN1   | 000-101b | 受信パケット長を設定します。XWDLEN1と等しくする必要があります。                | RCR1  | 2-16ページの表2-7 |
| XWDLEN1   | 000-101b | 送信パケット長を設定します。RWDLEN1と等しくする必要があります。                | XCR1  | 2-19ページの表2-9 |

### 2.7.2 SPIマスタとしてのMcBSP動作

McBSPがSPIマスタとして動作する場合、McBSPはシリアル・クロック信号を供給してデータの送信を制御します。BCLKXピン上のクロック信号は、パケット転送の間だけイネーブルになります。パケットが転送されていないときは、BCLKXピンは選択されている極性によってハイまたはローになります。McBSPをマスタとして使用するSPIインターフェイスの例を、図2-62に示しました。SPIマスタ動作では、BCLKXピンを出力として設定する必要があります。この場合、CPUクロックからBCLKX信号を生成するためにサンプル・レート生成器が使用されます。クロック・ストップ・モードでは、BCLKXピンがBCLKRピンに内部で接続されるため、BCLKRピンでは外部の信号接続が必要なく、送信回路と受信回路の両方がマスタ・クロック(BCLKX)によってクロック処理されます。

McBSPは、BFSXピンからスレーブ・イネーブル信号(SS)を供給することもできます。スレーブ・イネーブル信号が必要な場合は、BFSXピンを出力として設定し、パケットが送信されるたびにフレーム同期パルスが生成されるようフレーム生成器を設定する必要があります。このモードでは0または2というデータ遅延の値は定義されていないため、SPIマスタを正しく動作させるには、McBSPのデータ遅延パラメータ(XDATDLYおよびRDATDLY)を1に設定する必要があります。BFSXピンの極性はハイまたはローにプログラムできます。しかし、通常はBFSXピンをアクティブ・ローに設定します。

この設定では、サンプル・レート生成器レジスタのフレーム生成器ビット/フィールド(FPERおよびFWID)は無視され、フレーム同期波形の自由な設定はできません。結果、BFSXピンで生成される波形を、図2-64から図2-67までに示しました。信号はパケット転送の先頭ビットよりも前にアクティブになり、パケットの最終ビットが転送されるまでアクティブのままです。パケット転送が完了した後、BFSX信号がインアクティブ状態に戻ります。

複数のワードを連続して転送する場合でも、パケット転送の後BCLKX信号は停止され、BFSX信号はパケット転送の後インアクティブ状態に戻ることに注意してください。連続するパケット転送を実行すると、各パケット転送間に2ビット期間のアイドル時間が生じる結果となります。表2-31に、McBSPをマスタとして設定するために必要なレジスタ・ビット値を示します。

表2-31. SPIマスタ動作のレジスタ・ビット値

| ビット/フィールド | 値     | 説明                               | レジスタ  | 参照            |
|-----------|-------|----------------------------------|-------|---------------|
| CLKXM     | 1     | BCLKXピンを出力として設定します。              | PCR   | 2-12ページの表2-6  |
| CLKSM     | 1     | サンプル・レート・クロックがCPUクロックを用いて生成されます。 | SRGR2 | 2-60ページの表2-18 |
| CLKGDV    | 1-255 | サンプル・レート・クロックの分周率を決定します。         | SRGR1 | 2-59ページの表2-17 |
| FSXM      | 1     | BFSXピンを出力として設定します。               | PCR   | 2-12ページの表2-6  |
| FSGM      | 0     | 各パケット転送についてBFSX信号をアクティブにします。     | SRGR2 | 2-60ページの表2-18 |
| FSXP      | 1     | BFSXピンをアクティブ・ローに設定します。           | PCR   | 2-12ページの表2-6  |
| XDATDLY   | 01b   | BFSX信号の正しいセットアップ時間を設定します。        | XCR2  | 2-20ページの表2-10 |
| RDATDLY   | 01b   | BFSX信号の正しいセットアップ時間を設定します。        | RCR2  | 2-17ページの表2-8  |

## 2.7.3 SPIスレーブとしてのMcBSP動作

McBSPをSPIスレーブとして使用する場合、マスタ・クロックとスレーブ・イネーブル信号はマスタ・デバイスによって外部で生成されます。したがって、BCLKXピンとBFSXピンを入力として設定する必要があります。BCLKXピンはBCLKR信号に内部で接続されるため、McBSPの送信回路と受信回路の両方が外部マスタ・クロックによってクロック処理されます。BFSXピンもBFSR信号に内部で接続されるため、BCLKRピンとBFSRピンには外部との信号接続は必要ありません。McBSPをスレーブとして使用するSPIインターフェイスの例を、図2-63に示しました。

BCLKX信号はマスタによって外部で生成され、McBSPには同期しませんが、SPIスレーブとして正しく動作させるにはMcBSPのサンプル・レート生成器をイネーブルにする必要があります。サンプル・レート生成器は、最大周期の1/2CPUクロック周期に設定します。この場合、McBSPを外部のマスタ・クロックおよびスレーブ・イネーブル信号に同期させるために内部のサンプル・レート・クロックが使用されます。

McBSPは、転送のたびに、BFSXで入力されるスレーブ・イネーブル信号のアクティブ・エッジを必要とします。これは、マスタ・デバイスは各転送の先頭でスレーブ・イネーブル信号をアクティブにし、各パケット転送の完了後に信号をインアクティブにする必要があることを意味します。スレーブ・イネーブル信号は、転送が行われていないときにもアクティブのままにしておくことはできません。SPIスレーブとして正しく動作させるには、McBSPのデータ遅延パラメータを0に設定する必要があります。1または2という値はこの動作モードでは定義されていません。表2-32に、McBSPをスレーブとして設定するために必要なレジスタ・ビット値を示します。

表2-32. SPIスレーブ・オペレーション用のレジスタ・ビット値

| ビット/フィールド | 値 | 説明                               | レジスタ  | 参照            |
|-----------|---|----------------------------------|-------|---------------|
| CLKXM     | 0 | BCLKXピンを入力として設定します。              | PCR   | 2-12ページの表2-6  |
| CLKSM     | 1 | サンプル・レート・クロックがCPUクロックを用いて生成されます。 | SRGR2 | 2-60ページの表2-18 |
| CLKGDV    | 1 | サンプル・レート・クロックを2分周に設定します。         | SRGR1 | 2-59ページの表2-17 |
| FSXM      | 0 | BFSXピンを入力として設定します。               | PCR   | 2-12ページの表2-6  |
| FSGM      | 0 | 各パケット転送についてBFSX信号をアクティブにします。     | SRGR2 | 2-60ページの表2-18 |
| FSXP      | 1 | BFSXピンをアクティブ・ローに設定します。           | PCR   | 2-12ページの表2-6  |
| XDATDLY   | 0 | SPIスレーブ動作では0に設定する必要があります。        | XCR2  | 2-20ページの表2-10 |
| RDATDLY   | 0 | SPIスレーブ動作では0に設定する必要があります。        | RCR2  | 2-17ページの表2-8  |

## 2.7.4 SPIモードでのMcBSP初期化

デバイス・リセット、送信部リセット、または受信部リセット中のMcBSPの動作については、2-22ページの第2.3.1節「シリアル・ポートのリセット」で説明しました。SPIマスタまたはスレーブ動作用にMcBSPを設定するには、次の手順を実行する必要があります。

- 1) リアル・ポート制御レジスタ2(SPCR2)の送信部リセット・ビット( $\overline{\text{XRST}}$ )をゼロに設定して、送信部をリセットします。シリアル・ポート制御レジスタ1(SPCR1)の受信部リセット・ビット( $\overline{\text{RRST}}$ )をゼロに設定して、受信部をリセットします。

$$(\overline{\text{XRST}}=\overline{\text{RRST}}=0)$$

- 2) 表2-30のようにMcBSPレジスタ・フィールドをプログラムします。また、McBSPをマスタにするかスレーブにするかにより、表2-31または表2-32のようにレジスタ・フィールドをプログラムします。他のすべてのMcBSPレジスタ・フィールドは、デフォルト値にプログラムして構いません。

- 3) リアル・ポート制御レジスタ2(SPCR2)のサンプル・レート生成器リセット・ビット( $\overline{\text{GRST}}$ )を1に設定して、サンプル・レート生成器のリセットを解除します。SPCR2にライトする値では、 $\overline{\text{GRST}}$ ビットだけを1に変更し、残りのビット/フィールドは手順2でライトされた値と同じにします。

$$(\overline{\text{GRST}}=1)$$

- 4) McBSPが安定するまで、2サンプル・レート生成器クロック期間待ちます。
- 5) CPUまたはDMAコントローラのどちらがMcBSPのデータ転送を制御するかにより、手順(a)または(b)を実行します。CPUがMcBSPのデータ転送を制御する場合は手順(a)を、DMAコントローラがMcBSPのデータ転送を制御する場合は手順(b)を実行します。

- a) CPUがMcBSPのデータ転送を制御する場合、 $\overline{\text{XRST}}$ および $\overline{\text{RRST}}$ ビット/フィールドを1にセットして、送信部と受信部をイネーブルにします。SPCR1とSPCR2にライトするときは、リセット・ビットだけを1にセットし、残りのビット/フィールドは手順2でライトされた値と同じにします。

$$(\overline{\text{XRST}}=\overline{\text{RRST}}=1)$$

- b) DMAコントローラがMcBSPのデータ転送を制御する場合。最初に、DMAコントローラを設定してMcBSPのデータ転送を制御するチャンネルをイネーブルにします。次に、 $\overline{\text{XRST}}$ および $\overline{\text{RRST}}$ ビット/フィールドを1にセットして、送信部と受信部をイネーブルにします。SPCR1とSPCR2にライトするときは、リセット・ビットだけを1にセットし、残りのビット/フィールドは手順2でライトされた値と同じします。

$$(\overline{\text{XRST}}=\overline{\text{RRST}}=1)$$

- 6) McBSPロジックが安定するまで、サンプル・レート生成器の2クロック期間待ちます。



## 2.8 エミュレーションFREEビットおよびSOFTビット

FREEおよびSOFTは、高級言語デバッガにおいてブレークポイントに遭遇したときのシリアル・ポート・クロックの状態を決定する、特殊なエミュレーション・ビットです。FREEビットを1にセットすると、ソフトウェア・ブレークポイントに遭遇しても、クロックは動作を継続し(フリー・ラン)、データが引き続きシフト・アウトされます。FREE = 1の場合、SOFTビットは参照されません。FREEビットがゼロにクリアされると、SOFTビットが有効になります。SOFTビットがゼロにクリアされると、クロックが即座に停止し、したがって送信が中止されます。SOFTビットが1にセットされ、送信が途中であれば、転送が完了するまで送信が継続された後、クロックが停止します。表2-33にこれらの各設定を示します。

受信部側での機能も同様です。即時停止(SOFT = FREE = 0)以外のオプションを選択した場合、受信部は動作を継続するため、オーバーフロー・エラーが発生する可能性があることに注意してください。

表2-33. McBSPクロック設定

| FREE | SOFT | McBSPクロック設定                             |
|------|------|-----------------------------------------|
| 0    | 0    | 即時停止、クロックが停止します(リセット値)。                 |
| 0    | 1    | 現在のワードの送信が完了してから送信部が停止します。受信部は影響を受けません。 |
| 1    | 不特定  | フリー・ラン                                  |

## 2.9 McBSPの汎用入出力ポート

次の2つの条件が満たされれば、シリアル・ポート・ピン(CLKX、FSX、DX、CLKR、FSR、およびDR)をシリアル・ポートのピンではなく汎用入出力のピンとして使用できます。

- 1) シリアル・ポートの対応する部分(送信部または受信部)がリセット状態にある  
; SPCR[1,2]で $(R/X)\overline{RST} = 0$
- 2) シリアル・ポートの対応する部分について、汎用I/Oがイネーブルである; PCRで $(R/X)IOEN = 1$

2-12ページの図2-4「ピン制御レジスタ(PCR)」に、各McBSPピンを汎用入力または汎用出力として設定するためのビットを示しました。表2-34に、その設定方法を示します。FS(R/X)の場合、FS(R/X)M = 0ではピンが入力として設定され、FS(R/X)M = 1ではピンが出力として設定されます。出力として設定された場合、FS(R/X)Pにライトされた値がFS(R/X)に出力されます。入力として設定された場合、FS(R/X)Pはその信号のステータスを反映するリード専用ビットになります。CLK(R/X)MとCLK(R/X)Pも、CLK(R/X)に対して同様に機能します。送信部を汎用I/Oとして選択すると、PCRのDX\_STATビットの値がDXに出力されます。DRは常に入力であり、その値はPCRのDR\_STATビットに格納されます。CLKSは常にMcBSPへの入力であり、CLKSを汎用入力として設定するには、送信動作と受信動作の両方に影響を及ぼすため、送信部と受信部の両方をリセット状態にして $(R/X)IOEN = 1$ に設定する必要があります。

表2-34. 汎用I/Oとしてのピンの設定

| ピン   | 両方を設定して汎用I/O<br>をイネーブル                                                                     | 出力として選択            | 出力値のライト<br>用ビット | 入力として選択            | 入力値のリード<br>用ビット |
|------|--------------------------------------------------------------------------------------------|--------------------|-----------------|--------------------|-----------------|
| CLKX | $\overline{\text{XRST}} = 0$<br>$\text{XIOEN} = 1$                                         | $\text{CLKXM} = 1$ | CLKXP           | $\text{CLKXM} = 0$ | CLKXP           |
| FSX  | $\overline{\text{XRST}} = 0$<br>$\text{XIOEN} = 1$                                         | $\text{FSXM} = 1$  | FSXP            | $\text{FSXM} = 0$  | FSXP            |
| DX   | $\overline{\text{XRST}} = 0$<br>$\text{XIOEN} = 1$                                         | 常時                 | DX_STAT         | 不可                 | 適用外             |
| CLKR | $\overline{\text{RRST}} = 0$<br>$\text{RIOEN} = 1$                                         | $\text{CLKRM} = 1$ | CLKRP           | $\text{CLKRM} = 0$ | CLKRP           |
| FSR  | $\overline{\text{RRST}} = 0$<br>$\text{RIOEN} = 1$                                         | $\text{FSRM} = 1$  | FSRP            | $\text{FSRM} = 0$  | FSRP            |
| DR   | $\overline{\text{RRST}} = 0$<br>$\text{RIOEN} = 1$                                         | 不可                 | 適用外             | 常時                 | DR_STAT         |
| CLKS | $\overline{\text{RRST}} = \overline{\text{XRST}} = 0$<br>$\text{RIOEN} = \text{XIOEN} = 1$ | 不可                 | 適用外             | 常時                 | CLKS_STAT       |

## 2.10 パワーダウン・モードにおけるMcBSPの動作

‘C54xデバイスは、デバイスの全体または一部を休止状態にし、消費電力を通常の動作時よりも少なくできるパワーダウン・モードをいくつか備えています。パワーダウン・モードは、IDLE命令を実行したり、HMステータス・ビットを1にセットして $\overline{\text{HOLD}}$ 入力をローにドライブするなど、いくつかの方法で呼び出すことができます。McBSPは、他のペリフェラルと同様、送信割り込みまたは受信割り込みによってCPUのIDLEを解除できます。

IDLE1モードまたはHOLDモードのときは、McBSPは制限なく通常の動作を継続します。

IDLE2モードまたはIDLE3モードでは、ペリフェラルに供給される内部デバイス・クロックが停止します。その結果、これらのモードではMcBSPの動作にいくつかの制限が加えられます。外部からクロックとフレーム同期が供給される場合は、McBSPは動作を継続でき、受信割り込みと送信割り込みによってIDLE状態を解除できます。クロックまたはフレーム同期のどちらかが内部(クロックおよびフレーム同期生成器によって供給される)の場合は、McBSPが停止します。IDLE2/3ではクロックおよびフレーム同期生成器の動作に内部クロックを使用できないためです。

IDLE2/3では、転送が開始されるとMcBSPおよびDMAコントローラへの内部クロックが自動的に起動し、転送完了後に停止します。転送が行われないときにはどちらのモジュールも動作しないため、この機能によってさらに電力を節約できます。

自動バッファリング設定では、ハーフ・バッファまたはフル・バッファのデータが転送されるまでCPUをIDLE状態のままにして、McBSPおよびDMAを動作させることができます。各ワードの受信/送信時に割り込みが発生しないように、McBSPの受信割り込みと送信割り込みをディセーブルにします。DMAは自動バッファ(ABU)モードに設定し、DMA転送はMcBSPの受信イベントおよび送信イベント時に行われるように設定します。この場合DMAは、ハーフ・バッファまたはフル・バッファのうちいずれか希望する方の転送完了時にCPUへの割り込みを実行するよう設定します。CPUは割り込みが発生するまでIDLE状態のままであるため、CPUのIDLE状態を終了せずにデータ・ブロック全体を転送することができます。DMAの動作の詳細については、第3章「ダイレクト・メモリー・アクセス(DMA)コントローラ」を参照してください。

## 2.11 McBSPプログラミングのサンプル・プログラム

第3章に、McBSPプログラミングのサンプル・プログラムがあります: 例3-19「ABUモードでのMcBSPデータ転送」は3-52ページに、例3-20「ダブル・ワード・モードでのMcBSPデータ転送」は3-54ページに、例3-21「McBSPからデータ・メモリーへの転送(データ・ソートあり)」は3-56ページに記載されています。

2



# ダイレクト・メモリー・アクセス(DMA)コントローラ

DMA(Direct Memory Access)コントローラは、いくつかの'C54x製品上でのみ機能します。本章では、'C5402、'C5410、'C5420で利用できるDMAの動作と設定方法について説明します。

| Topic                                              | Page |
|----------------------------------------------------|------|
| 3.1 DMAの概要 .....                                   | 3-2  |
| 3.2 DMAの動作および設定方法 .....                            | 3-4  |
| 3.3 拡張アドレッシング .....                                | 3-33 |
| 3.4 DMAのメモリー・マップ .....                             | 3-34 |
| 3.5 DMAの転送レイテンシー .....                             | 3-39 |
| 3.6 DMAコントローラを介した拡張ホスト・ポート・インターフェイスの<br>アクセス ..... | 3-42 |
| 3.7 'C5420の内部でのプロセッサ間FIFOによる通信 .....               | 3-43 |
| 3.8 パワーダウン・モードでのDMA動作 .....                        | 3-43 |
| 3.9 プログラム例 .....                                   | 3-44 |

### 3.1 DMAの概要

DMA(Direct Memory Access)コントローラは、CPUと独立してメモリー・マップ領域内でデータ転送を行います。DMAを使用すれば、内部メモリー、内部ペリフェラル、または外部デバイス間のデータ転送をCPU動作と独立して行うことができます。DMAには、個々にプログラム可能な6つのチャンネルが用意されており、6種類の動作設定が可能です。また、DMAコントローラはホスト・ポート・インターフェイス(HPI-8/HPI-16)からの要求に対してDMAバスを提供します。なお、この章全体を通して、HPI-8とHPI-16のどちらを述べる場合にもHPIxという省略形を使用します。

DMA動作の説明に使われる用語を次のように定義します。

- リード転送 DMAがメモリー内の転送元アドレス位置からデータをリードします。転送元はメモリーかペリフェラルで、プログラム、データ、またはI/O空間内に配置されています。
- ライト転送 DMAがリード転送時にリードしたデータを、メモリー内の転送先アドレス位置にライトします。転送先はメモリーかペリフェラルで、プログラム、データ、またはI/O空間内に配置されています。
- エレメント転送 1つのデータ・エレメントに対するリード／ライト転送です。
- フレーム転送 DMAの各チャンネルで、1つのフレームにつき転送するエレメント数を個別にプログラムすることができます。DMAが1フレーム内のすべてのエレメントを転送すると、フレーム転送が完了します。
- ブロック転送 DMAの各チャンネルで、1つのブロックにつき転送するフレーム数を個別にプログラムすることができます。DMAがブロックに対して設定されたすべてのフレームを転送すると、ブロック転送が完了します。

DMAの特長を次に示します。

- バックグラウンド動作 DMAはCPUと独立して動作します。
- 6チャンネル DMAは、6つの独立したブロック転送を設定できます。
- ホスト・ポート・インターフェイスによるアクセス DMAは、CPUと独立してホスト・ポート・インターフェイスのデータ用にメモリーへのバスを提供します。これにより、ホスト・ポート・インターフェイスはこれまで以上に多くのメモリー空間(オンチップ・メモリー全体)にアクセスできるようになります。
- マルチフレーム転送 各ブロック転送では、サイズをプログラムできるようなマルチフレームを構成することができます。詳細については、3-13ページの第3.2.3.3節「DMA同期イベントおよびフレーム・カウント・レジスタ」を参照してください。
- 優先順位をプログラム可能 優先順位の高低をチャンネルごとに設定することができます。詳細については、3-7ページの第3.2.2節「DMAチャンネル優先順位およびイネーブル制御レジスタ」を参照してください。



- アドレスをプログラム可能 各チャネルの転送元と転送先のアドレス・レジスタは、リード／ライト転送に対してインデックスを設定することができます。アドレスは、定数、インクリメント/デクリメント、あるいはプログラムした値で変更することも可能です。プログラムした値を使用すれば、フレーム内の最後の転送とそれ以前の転送に対してそれぞれ異ったインデックスを用いる事ができます。また、データをソートすることも可能です(これについては、本章で後述します)。
- アドレス範囲 DMAでは、デバイス上の拡張アドレス範囲全域にアクセスすることができます。DMAでアクセスできる範囲を次に示します。
  - オンチップ・メモリー
  - オンチップ・ペリフェラル
  - 外部メモリー(特定デバイス)。メモリー・マップでアクセスできる領域はデバイスによって異なります。詳細については、ユーザーズ・ガイド『TMS320C54x DSP, CPU and Peripherals, Reference Set Volume 1』(文献番号SPRU131)の第3章「Memory」を参照してください。
- プログラム可能な転送幅 チャネルごとにシングル・ワード・モード(16ビット)かダブル・ワード・モード(32ビット)のどちらで転送するかを設定できます。詳細については、3-13ページの第3.2.3.3節「DMA同期イベントおよびフレーム・カウント・レジスタ」を参照してください。
- 自動初期化 ブロック転送が完了した後、次のブロック転送のためにDMAチャネルを自動的に再初期化できます。詳細については、3-17ページの第3.2.3.4節「転送モード制御レジスタ」を参照してください。
- 同期イベント 選択したイベントによって各エレメント転送を開始することができます。詳細については、3-13ページの第3.2.3.3章「DMA同期イベントおよびフレーム・カウント・レジスタ」を参照してください。
- 割り込み発生 フレーム転送またはブロック転送が完了するごとに、各DMAチャネルがCPUに割り込みをかけられます。詳細については、3-9ページの第3.2.2.2節「多重割り込み制御」を参照してください。

## 3.2 DMAの動作および設定方法

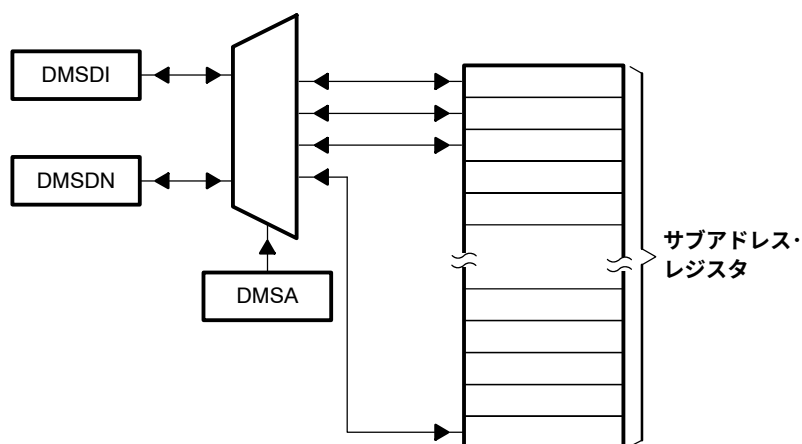
DMAの動作と設定には、メモリーにマッピングされた複数の制御レジスタが用いられます。この方法は他の'C54xデバイスのペリフェラルで用いられてるものと似ています。DMAレジスタはサブアドレッシングという方法でメモリーにマッピングされています。

### 3

#### 3.2.1 レジスタのサブアドレッシング

レジスタのサブアドレッシングでは、1つのメモリー・マップ内に複数のレジスタ(サブアドレス・レジスタ)をマルチプレクスしています。マルチプレクサの制御には、サブバンク・アドレス・レジスタが使われます。希望のサブアドレス・レジスタに実際にデータをリード/ライトするにはサブバンク・データ・レジスタが使われます。このしくみを具体的に表したものが図3-1です。こうすれば、多数のレジスタを少ないメモリー空間内にマッピングできます。

図3-1. レジスタのサブアドレッシング



特定のサブアドレス・レジスタにアクセスする時は、まず希望のサブアドレスをサブバンク・アドレス・レジスタ(DMSA)にライトします。これで、サブバンク・アクセス・レジスタ(DMSDIとDMSDN)と希望するアドレス位置を結ぶように、マルチプレクサに指示を出します。そして、サブバンク・データ・レジスタにアクセス(リード/ライト)すれば、サブバンク・アドレス・レジスタで事前に設定したように、希望のデータ・レジスタとの間でデータがやり取りされます。

DMAコントローラには2種類のサブバンク・アクセス・レジスタがあります。ひとつはDMSDIで、サブアドレスの自動インクリメント機能を持つサブバンク・アクセス・レジスタです。つまり、このレジスタではアクセスが終了するたびに、サブアドレスを自動的にインクリメントします。DMA用の制御レジスタ全てを設定する場合にDMSDIを使用すると、サブバンク・アドレスを1度設定すればレジスタ全てを設定するまで、アクセスごとにアドレスをポストインクリメントしてくれます。レジスタのアクセスが1回だけの時や、サブアドレスを変更しない場合には、DMSDNを使います。DMSDNを使用すると、サブアドレスは変更せずにサブアドレス・レジスタにアクセスできます。以上の方法で、DMAのプログラミングに使われるレジスタ全てを、4つのメモリー・マップド・レジスタに割り当てています。

例3-1と例3-2にサブアドレッシングの使用例を示します。

#### 例3-1. サブアドレッシング使用例(自動インクリメントなし)

次のサンプル・プログラムは、DMAチャネル5の転送元アドレス・レジスタを1000hに初期化する例を示しています。

```
DMSA      .set    55h          ;subbank address register location
DMSDN      .set    57h
DMSRC5     .set    19h
STM        DMSRC5,DMSA        ;initialize the subbank address
                                ;register to point to DMSRC5
STM        #1000h,DMSDN       ;write 1000h to DMSRC5
```

#### 例3-2. サブアドレッシング使用例(自動インクリメントあり)

次のサンプル・プログラムは、DMAチャネル5のコンテキスト・レジスタをDMSDIを用いて初期化する例を示しています。この例でレジスタに書き込まれる値は任意の値であり、特定の設定を表したものではありません。

```
DMSA      .set    55h          ;set register locations
DMSDI      .set    56h
DMSRC5     .set    19h
DMDST5     .set    1Ah
DMCTR5     .set    1Bh
DMSFC5     .set    1Ch
DMMCR5     .set    1Dh
STM        DMSRC5,DMSA        ;initialize the subbank address
                                ;register to point to DMSRC5
STM        #1000h,DMSDI       ;write 1000h to DMSRC5
```

## 例3-2 サブアドレッシング使用例(自動インクリメントあり)(続き)

```

STM      #2000h,DMSDI      ;write 2000h to DMDST5
STM      #0010h,DMSDI      ;write 10h to DMCTR5
STM      #0002h,DMSDI      ;write 2h to DMSFC5
STM      #0000h,DMSDI      ;write 0h to DMMCR5

```

表3-1に各種のDMAレジスタを示します。アドレスを直接指定するのは、チャンネル優先順位およびイネーブル制御レジスタ(DMPREC)のみです。その他のDMAレジスタはすべてサブアドレスで指定します。

表3-1. DMAレジスタ

| アドレス | サブアドレス | 名前     | 機能                            |
|------|--------|--------|-------------------------------|
| 54h  | —      | DMPREC | チャンネル優先順位およびイネーブル制御レジスタ       |
| 55h  | —      | DMSA   | サブバンク・アドレス・レジスタ               |
| 56h  | —      | DMSDI  | サブバンク・アクセス・レジスタ(自動インクリメントあり)  |
| 57h  | —      | DMSDN  | サブバンク・アクセス・レジスタ(自動インクリメントなし)  |
| —    | 00h    | DMSRC0 | チャンネル0転送元アドレス・レジスタ            |
| —    | 01h    | DMDST0 | チャンネル0転送先アドレス・レジスタ            |
| —    | 02h    | DMCTR0 | チャンネル0エレメント・カウント・レジスタ         |
| —    | 03h    | DMSFC0 | チャンネル0同期イベントおよびフレーム・カウント・レジスタ |
| —    | 04h    | DMMCR0 | チャンネル0転送モード制御レジスタ             |
| —    | 05h    | DMSRC1 | チャンネル1転送元アドレス・レジスタ            |
| —    | 06h    | DMDST1 | チャンネル1転送先アドレス・レジスタ            |
| —    | 07h    | DMCTR1 | チャンネル1エレメント・カウント・レジスタ         |
| —    | 08h    | DMSFC1 | チャンネル1同期イベントおよびフレーム・カウント・レジスタ |
| —    | 09h    | DMMCR1 | チャンネル1転送モード制御レジスタ             |
| —    | 0Ah    | DMSRC2 | チャンネル2転送元アドレス・レジスタ            |
| —    | 0Bh    | DMDST2 | チャンネル2転送先アドレス・レジスタ            |
| —    | 0Ch    | DMCTR2 | チャンネル2エレメント・カウント・レジスタ         |
| —    | 0Dh    | DMSFC2 | チャンネル2同期イベントおよびフレーム・カウント・レジスタ |
| —    | 0Eh    | DMMCR2 | チャンネル2転送モード制御レジスタ             |
| —    | 0Fh    | DMSRC3 | チャンネル3転送元アドレス・レジスタ            |
| —    | 10h    | DMDST3 | チャンネル3転送先アドレス・レジスタ            |
| —    | 11h    | DMCTR3 | チャンネル3エレメント・カウント・レジスタ         |
| —    | 12h    | DMSFC3 | チャンネル3同期イベントおよびフレーム・カウント・レジスタ |
| —    | 13h    | DMMCR3 | チャンネル3転送モード制御レジスタ             |
| —    | 14h    | DMSRC4 | チャンネル4転送元アドレス・レジスタ            |
| —    | 15h    | DMDST4 | チャンネル4転送先アドレス・レジスタ            |

表3-1. DMAレジスタ (続き)

| アドレス | サブアドレス | 名前     | 機能                            |
|------|--------|--------|-------------------------------|
| —    | 16h    | DMCTR4 | チャンネル4エレメント・カウント・レジスタ         |
| —    | 17h    | DMSFC4 | チャンネル4同期イベントおよびフレーム・カウント・レジスタ |
| —    | 18h    | DMMCR4 | チャンネル4転送モード制御レジスタ             |
| —    | 19h    | DMSRC5 | チャンネル5転送元アドレス・レジスタ            |
| —    | 1Ah    | DMDST5 | チャンネル5転送先アドレス・レジスタ            |
| —    | 1Bh    | DMCTR5 | チャンネル5エレメント・カウント・レジスタ         |
| —    | 1Ch    | DMSFC5 | チャンネル5同期イベントおよびフレーム・カウント・レジスタ |
| —    | 1Dh    | DMMCR5 | チャンネル5転送モード制御レジスタ             |
| —    | 1Eh    | DMSRCP | 転送元プログラム・ページ・アドレス(全チャンネル)     |
| —    | 1Fh    | DMDSTP | 転送先プログラム・ページ・アドレス(全チャンネル)     |
| —    | 20h    | DMIDX0 | エレメント・アドレス・インデックス・レジスタ0       |
| —    | 21h    | DMIDX1 | エレメント・アドレス・インデックス・レジスタ1       |
| —    | 22h    | DMFRI0 | フレーム・アドレス・インデックス・レジスタ0        |
| —    | 23h    | DMFRI1 | フレーム・アドレス・インデックス・レジスタ1        |
| —    | 24h    | DMGSA  | グローバル転送元アドレス・リロード・レジスタ        |
| —    | 25h    | DMGDA  | グローバル転送先アドレス・リロード・レジスタ        |
| —    | 26h    | DMGCR  | グローバル・エレメント・カウント・リロード・レジスタ    |
| —    | 27h    | DMGFR  | グローバル・フレーム・カウント・リロード・レジスタ     |

3

### 3.2.2 DMAチャンネル優先順位およびイネーブル制御レジスタ

チャンネル優先順位およびイネーブル制御(DMPREC)レジスタは、DMAシステム全体の動作に関するいくつかの機能を制御します。以下にその制御内容を示します。

- ☐ 各DMAチャンネルを選択してイネーブル
- ☐ 多重割り込みを制御
- ☐ チャンネルの優先順位を制御

図3-2に、DMPRECの構成を示します。DMPRECはデータ空間の0054h番地に配置されており、サブアドレスされません。リセットを行うと、DMPRECは0000hに初期化されます。

図3-2. DMAチャンネル優先順位およびイネーブル制御(DMPREC)レジスタ

|      |      |      |         |         |   |   |   |
|------|------|------|---------|---------|---|---|---|
| 15   | 14   | 13   | 8       | 7       | 6 | 5 | 0 |
| FREE | RSVD | DPRC | INTOSEL | DE[5:0] |   |   |   |

表3-2. DMAチャネル優先順位およびイネーブル制御(DMPREC)レジスタのビット/フィールド説明

| ビット | 名前      | リセット値 | 機能                                                                                                                   |
|-----|---------|-------|----------------------------------------------------------------------------------------------------------------------|
| 15  | FREE    | 0     | このビットはエミュレーション時のDMAコントローラの動作を制御します。FREE=0の場合は、エミュレータ停止時にDMA転送は即停止します。FREE=1の場合は、エミュレーションが停止してもDMA転送は続行されます。          |
| 14  | RSVD    | 0     | 予約。このフィールドにライトされた値は無効です。                                                                                             |
| 13  | DPRC[5] | 0     | DMAチャネル5優先順位制御ビット<br>DPRC[5] = 1 高優先順位<br>DPRC[5] = 0 低優先順位                                                          |
| 12  | DPRC[4] | 0     | DMAチャネル4優先順位制御ビット<br>DPRC[4] = 1 高優先順位<br>DPRC[4] = 0 低優先順位                                                          |
| 11  | DPRC[3] | 0     | DMAチャネル3優先順位制御ビット<br>DPRC[3] = 1 高優先順位<br>DPRC[3] = 0 低優先順位                                                          |
| 10  | DPRC[2] | 0     | DMAチャネル2優先順位制御ビット<br>DPRC[2] = 1 高優先順位<br>DPRC[2] = 0 低優先順位                                                          |
| 9   | DPRC[1] | 0     | DMAチャネル1優先順位制御ビット<br>DPRC[1] = 1 高優先順位<br>DPRC[1] = 0 低優先順位                                                          |
| 8   | DPRC[0] | 0     | DMAチャネル0優先順位制御ビット<br>DPRC[0] = 1 高優先順位<br>DPRC[0] = 0 低優先順位                                                          |
| 7-6 | INTOSEL | 0     | 割り込み多重制御ビット。INTOSELビットは、割り込みベクタ・テーブルとIMR/IMFレジスタで、DMA割り込みの割り当ての設定を行います。このフィールドの設定は、デバイスによって異なります(表3-3、表3-4、表3-5を参照)。 |
| 5   | DE[5]   | 0     | DMAチャネル5イネーブル・ビット<br>DE[5] = 1 DMAチャネル5をイネーブル<br>DE[5] = 0 DMAチャネル5をディセーブル                                           |
| 4   | DE[4]   | 0     | DMAチャネル4イネーブル・ビット<br>DE[4] = 1 DMAチャネル4をイネーブル<br>DE[4] = 0 DMAチャネル4をディセーブル                                           |

表3-2. DMAチャンネル優先順位およびイネーブル制御(DMPREC)レジスタのビット/フィールド説明  
(続き)

| ビット | 名前    | リセット<br>値 | 機能                                                                            |
|-----|-------|-----------|-------------------------------------------------------------------------------|
| 3   | DE[3] | 0         | DMAチャンネル3イネーブル・ビット<br>DE[3] = 1 DMAチャンネル3をイネーブル<br>DE[3] = 0 DMAチャンネル3をディセーブル |
| 2   | DE[2] | 0         | DMAチャンネル2イネーブル・ビット<br>DE[2] = 1 DMAチャンネル2をイネーブル<br>DE[2] = 0 DMAチャンネル2をディセーブル |
| 1   | DE[1] | 0         | DMAチャンネル1イネーブル・ビット<br>DE[1] = 1 DMAチャンネル1をイネーブル<br>DE[1] = 0 DMAチャンネル1をディセーブル |
| 0   | DE[0] | 0         | DMAチャンネル0イネーブル・ビット<br>DE[0] = 1 DMAチャンネル0をイネーブル<br>DE[0] = 0 DMAチャンネル0をディセーブル |

## 3.2.2.1 DMAチャンネル・イネーブル制御

6つの各DMAチャンネルは、DMPRECのDEフィールドを使って個別にイネーブルにできます。このレジスタのビット0～5は、6つの各DMAチャンネルに対応しています(ビット0=チャンネル0、ビット1=チャンネル1…。)。各ビットが1の場合は対応するチャンネルがイネーブルになり、0の場合はディセーブルになります。

ブロック転送の完了時に、DMAコントローラは各イネーブル・ビットをリセットすることができます。また、DMPRECをポーリングすることで、特定チャンネルでのブロック転送が完了するタイミングを判断できます。DMAコントローラとCPUの両方がDEフィールドのビット状態を変更しようとした場合は、DMAコントローラが優先されます。

## 3.2.2.2 多重割り込み制御

DMAイベントにより、転送完了時にCPUへの割り込みを発生させることができます。'C54xメモリー・マップでは割り込みの数が限られているため、いくつかのDMA割り込みは他のペリフェラル割り込みとマルチプレクスされています。この割り込み機能は、DMPREC内のINTOSELフィールド(ビット7と6)によって制御されます。INTOSELフィールドでは、最大6つのDMA用の割り込みを設定することができます(設定方法はデバイスによって異なります)。

表3-3、表3-4、表3-5に、各デバイスで可能なDMA割り込みの割り当てを示します。リセット時、割り込み値は各表の00b列に示す値に設定されます。

表3-3. 'C5402での多重割り込みの割り当て

| 割り込み番号<br>(IMR/IFR #) | 00b          | INTOSEL[1:0]の値<br>01b | 10b           | 11b |
|-----------------------|--------------|-----------------------|---------------|-----|
| 7                     | タイマ1割り込み     | タイマ1割り込み              | DMAチャンネル1割り込み | 予約  |
| 10                    | McBSP 1 RINT | DMAチャンネル2割り込み         | DMAチャンネル2割り込み | 予約  |
| 11                    | McBSP 1 XINT | DMAチャンネル3割り込み         | DMAチャンネル3割り込み | 予約  |

表3-4. 'C5410での多重割り込みの割り当て

| 割り込み番号<br>(IMR/IFR #) | 00b           | INTOSEL[1:0]の値<br>01b | 10b           | 11b |
|-----------------------|---------------|-----------------------|---------------|-----|
| 4                     | McBSP 0 RINT  | McBSP 0 RINT          | McBSP 0 RINT  | 予約  |
| 5                     | McBSP 0 XINT  | McBSP 0 XINT          | McBSP 0 XINT  | 予約  |
| 6                     | McBSP 2 RINT  | McBSP 2 RINT          | DMAチャンネル0割り込み | 予約  |
| 7                     | McBSP 2 XINT  | McBSP 2 XINT          | DMAチャンネル1割り込み | 予約  |
| 10                    | McBSP 1 RINT  | DMAチャンネル2割り込み         | DMAチャンネル2割り込み | 予約  |
| 11                    | McBSP 1 XINT  | DMAチャンネル3割り込み         | DMAチャンネル3割り込み | 予約  |
| 12                    | DMAチャンネル4割り込み | DMAチャンネル4割り込み         | DMAチャンネル4割り込み | 予約  |
| 13                    | DMAチャンネル5割り込み | DMAチャンネル5割り込み         | DMAチャンネル5割り込み | 予約  |

表3-5. 'C5420での多重割り込みの割り当て(各サブシステム)

| 割り込み番号<br>(IMR/IFR #) | 00b           | INTOSEL[1:0]の値<br>01b | 10b           | 11b |
|-----------------------|---------------|-----------------------|---------------|-----|
| 4                     | McBSP 0 RINT  | McBSP 0 RINT          | McBSP 0 RINT  | 予約  |
| 5                     | McBSP 0 XINT  | McBSP 0 XINT          | McBSP 0 XINT  | 予約  |
| 6                     | McBSP 2 RINT  | McBSP 2 RINT          | DMAチャンネル0割り込み | 予約  |
| 7                     | McBSP 2 XINT  | McBSP 2 XINT          | DMAチャンネル1割り込み | 予約  |
| 10                    | McBSP 1 RINT  | DMAチャンネル2割り込み         | DMAチャンネル2割り込み | 予約  |
| 11                    | McBSP 1 XINT  | DMAチャンネル3割り込み         | DMAチャンネル3割り込み | 予約  |
| 12                    | DMAチャンネル4割り込み | DMAチャンネル4割り込み         | DMAチャンネル4割り込み | 予約  |
| 13                    | DMAチャンネル5割り込み | DMAチャンネル5割り込み         | DMAチャンネル5割り込み | 予約  |



### 3.2.2.3 DMAチャンネル優先順位制御

各DMAチャンネルは、低優先順位か高優先順位を個別に割り当てることができます。高優先順位のDMAチャンネルは、必ず低優先順位のチャンネルよりも先に処理されます。複数のチャンネルがイネーブルになっていて、それらが同じ優先順位に割り当てられている場合は、イネーブルになっているチャンネルが各チャンネル1エレメントずつ順次処理されます。

各チャンネルの優先順位は、DMPRECのDPRCフィールド(6ビット)で設定されます。各ビットがそれぞれのDMAチャンネルに対応しています。ビット8～13が、チャンネル0～5にそれぞれ対応します。ビットが0の場合は、対応するチャンネルに低優先順位が割り当てられ、ビットが1の場合は高優先順位が割り当てられます。

### 3.2.2.4 DMA機能に対するエミュレーション制御

DMPRECのFREEフィールドは、エミュレータが停止したときのDMAコントローラの動作を制御します。FREEが0にクリアされている場合は、エミュレータの停止時にDMA転送が即停止します。FREEが1にセットされている場合は、エミュレータが停止してもDMA転送は続行します。

### 3.2.3 チャンネル・コンテキスト・レジスタ

各DMAチャンネルには、5つのチャンネル・コンテキスト・レジスタがあり、これらのレジスタは各チャンネル専用の動作設定をします。それらの5つのレジスタを示します。

- ☐ 転送元アドレス・レジスタ
- ☐ 転送先アドレス・レジスタ
- ☐ エレメント・カウント・レジスタ
- ☐ 同期イベント/フレーム・カウント・レジスタ
- ☐ 転送モード制御レジスタ

各レジスタの動作とその影響については、以降の節で詳しく説明します。

#### 3.2.3.1 転送元アドレス・レジスタと転送先アドレス・レジスタ

DMAの各エレメント転送では、リードとそれに続くライトが必要になります。リードされるデータのアドレスを転送元アドレスといい、データのライト先となるアドレスを転送先アドレスといいます。各チャンネルにつき2つのチャンネル・コンテキスト・レジスタ(DMAチャンネル $n$ 転送元アドレス・レジスタ(DMSRC $n$ )とDMAチャンネル $n$ 転送先アドレス・レジスタ(DMDST $n$ ))があり、一方には転送元アドレスが、もう一方には転送先アドレスがストアされます。これらはどちらも16ビット・レジスタで、転送元または転送先を示す拡張アドレスの下位16ビットをストアします。

転送元アドレス・レジスタと転送先アドレス・レジスタは、DMA転送が始まる前にソフトウェアで初期化され、転送中はDMAコントローラによって自動的に更新されます。CPUでは必要に応じて、転送中に一方のアドレスを読み取り、転送状況をモニタすることができます。また、転送時にCPUを使ってこれらのレジスタにライトすることも可能です。ブロック転送中にアドレス・レジスタが変更されると、その変更は直ちに有効になり、その後の転送処理に影響します。転送中にこれらのレジスタに不必要なライトを行わないよう注意してください。

各チャンネルの転送元アドレス・レジスタと転送先アドレス・レジスタの配置については、表3-1を参照してください。

### 3.2.3.2 エレメント・カウント・レジスタ

各DMAチャンネルには、DMA転送の実行回数をカウントする16ビットのエレメント・カウンタがあります。このエレメント・カウンタは、DMAチャンネル・エレメント・カウント・レジスタ(DMCTRN)にストアされている符号なし16ビット値で初期化されます。この数値は転送されるエレメント数を表しています。エレメント・カウント・レジスタは、希望のエレメント転送数よりも1少ない値で初期化します。たとえば、DMAチャンネル2で9のデータ・エレメントを転送する場合は、DMCTR2は8で初期化します。

通常のマルチフレーム転送モードでは、DMAコントローラによって各転送ごとにDMCTRNが自動的にデクリメントされます。各フレーム内の最終エレメントに到達すると、エレメント・カウンタはシャドウ・レジスタにストアされているDMCTRNの値でリロードされます。

自動バッファリング(ABU)モードの場合は、DMCTRNのデータがバッファ・サイズを表します。このモードでは、転送中にDMCTRNがデクリメントされることはありません。ABUモードはDMAコントローラに組み込まれており、これによりC54xのバッファド・シリアル・ポート(BSP)の自動バッファリング・ユニットと同じ機能を持ちます。ABUモードのDMAとMcBSPを組み合わせれば、BSPのABUオペレーションと同等の動作が実行できます。ABUモードの詳細については、3-21ページの第3.2.3.5節「アドレッシング・モード」を参照してください。

CPUとDMAコントローラが同時にエレメント・カウント・レジスタを変更しようとした場合は、CPUが優先されます。

各チャンネルのエレメント・カウント・レジスタの配置については、表3-1を参照してください。

### 3.2.3.3 DMA同期イベントおよびフレーム・カウント・レジスタ

DMA同期イベントおよびフレーム・カウント・レジスタ(DMSFCn)には、次の3つの目的があります。

- ☐ DMA転送を開始するための同期イベントを決定する
- ☐ 転送ワード・サイズ(16ビットまたは32ビット)を決定する
- ☐ 転送フレーム数を決定する

3

図3-3に、DMSFCnの構成を示します。各チャネルのDMA同期イベントおよびフレーム・カウント・レジスタの配置については、表3-1を参照してください。

図3-3. DMA同期イベントおよびフレーム・カウント(DMSFCn)レジスタ

|           |    |      |      |   |             |   |
|-----------|----|------|------|---|-------------|---|
| 15        | 12 | 11   | 10   | 8 | 7           | 0 |
| DSYN[3:0] |    | DBLW | rsvd |   | Frame Count |   |

表3-6. DMA同期イベントおよびフレーム・カウント(DMSFCn)レジスタのビット/フィールド説明

| ビット   | 名前          | リセット値 | 機能                                                                                              |
|-------|-------------|-------|-------------------------------------------------------------------------------------------------|
| 15-12 | DSYN[3:0]   | 0     | DMA同期イベント。対応するDMAチャネルごとのDMA転送を開始するための同期イベントを指定します。使用可能な同期イベントについては、この表の後の「DMA同期イベント」節を参照してください。 |
| 11    | DBLW        | 0     | ダブル・ワード・モード<br>DBLW = 0 シングル・ワード・モード。各エレメントは16ビット。<br>DBLW = 1 ダブル・ワード・モード。各エレメントは32ビット。        |
| 10-8  | rsvd        | 0     | 予約。このフィールドにライトされた値は無効です。                                                                        |
| 7-0   | Frame Count | 0     | フレーム・カウント。総転送フレーム数を指定します。詳細については、表3-9の後の「フレーム・カウント」節を参照してください。                                  |

### DMA同期イベント

DMAのエレメント転送は、McBSPの受信/送信イベント、タイマ割り込みイベント、外部割り込みイベントなど、さまざまなイベントによって開始することができます。また、DMAチャネルを非同期で、どのイベントにも関係なく実行することも可能です。DMAチャネルが特定のイベントと同期されると、各エレメント転送はそのイベントが発生するのを待ちます。DMAチャネルがどのイベントとも同期されていないと、転送は可能な限り高速で処理されます。

同期イベントが用いられる場合、1回の転送に対して1つの同期イベントが必要になります。シングル・ワード・モードでは、1つの同期イベントによって16ビットのシングル・ワード転送が1回行われます。ダブル・ワード・モードの場合は、1つの同期イベントによって32ビットのダブル・ワード転送が1回行われます(16ビット転送を2回行うのと同じです)。

使用可能な同期イベントの種類は、'C54xデバイスによって異なります。前述した同期イベントは全てをどのデバイスでも使用できるとは限りません。表3-7、表3-8、表3-9に、各デバイスで使用可能な同期イベントを示します。これらの表に示されている適切な値をDMSFCnのDSYNフィールドにロードすることで、同期イベントが選択できます。それぞれのDMAチャンネルに対して選択できる同期イベントは1つだけです。リセットを行うと、DSYNフィールドは0000h(同期イベントなし)に設定されます。

表3-7. 'C5402でのDMA同期イベント

| DSYN[3:0]の値 | DMA同期モード            |
|-------------|---------------------|
| 0000        | 同期イベントなし(非同期動作)     |
| 0001        | McBSP0受信イベント(REVT0) |
| 0010        | McBSP0送信イベント(XEVT0) |
| 0011        | 予約                  |
| 0100        | 予約                  |
| 0101        | McBSP1受信イベント(REVT1) |
| 0110        | McBSP1送信イベント(XEVT1) |
| 0111        | 予約                  |
| 1000        | 予約                  |
| 1001        | 予約                  |
| 1010        | 予約                  |
| 1011        | 予約                  |
| 1100        | 予約                  |
| 1101        | タイマ0割り込みイベント        |
| 1110        | 外部割り込み3(INT3)イベント   |
| 1111        | タイマ1割り込みイベント        |

表3-8. 'C5410でのDMA同期イベント

| DSYN[3:0]の値 | DMA同期モード                       |
|-------------|--------------------------------|
| 0000        | 同期イベントなし(非同期動作)                |
| 0001        | McBSP0受信イベント(REVT0)            |
| 0010        | McBSP0送信イベント(XEVT0)            |
| 0011        | McBSP2受信イベント(REVT2)            |
| 0100        | McBSP2送信イベント(XEVT2)            |
| 0101        | McBSP1受信イベント(REVT1)            |
| 0110        | McBSP1送信イベント(XEVT1)            |
| 0111        | McBSP0受信イベント – ABISモード(REVTA0) |
| 1000        | McBSP0送信イベント – ABISモード(XEVTA0) |
| 1001        | McBSP2受信イベント – ABISモード(REVTA2) |
| 1010        | McBSP2送信イベント – ABISモード(XEVTA2) |
| 1011        | McBSP1受信イベント – ABISモード(REVTA1) |
| 1100        | McBSP1送信イベント – ABISモード(XEVTA1) |
| 1101        | タイマ割り込みイベント                    |
| 1110        | 外部割り込み3(INT3)イベント              |
| 1111        | 予約                             |

表3-9. 'C5420でのDMA同期イベント(各サブシステム)

| DSYN[3:0]の値 | DMA同期モード              |
|-------------|-----------------------|
| 0000        | 同期イベントなし(非同期動作)       |
| 0001        | McBSP0受信イベント(REVT0)   |
| 0010        | McBSP0送信イベント(XEVT0)   |
| 0011        | McBSP2受信イベント(REVT2)   |
| 0100        | McBSP2送信イベント(XEVT2)   |
| 0101        | McBSP1受信イベント(REVT1)   |
| 0110        | McBSP1送信イベント(XEVT1)   |
| 0111        | FIFO受信バッファのエンプティ・イベント |
| 1000        | FIFO送信バッファのフル・イベント    |
| 1001        | 予約                    |
| 1010        | 予約                    |
| 1011        | 予約                    |
| 1100        | 予約                    |
| 1101        | 予約                    |
| 1110        | 予約                    |
| 1111        | 予約                    |

### ダブル・ワード・モード

DMA転送は、シングル・ワードの16ビット転送か、ダブル・ワードの32ビット転送として設定できます。ダブル・ワード・モードでは、各32ビット転送が1つのエレメントと見なされます。ダブル・ワード転送では、16ビット転送が2回続けて行われ、この転送はDMAコントローラによって管理されており(適切なアドレス変更を含む)、CPUによる介入は一切ありません。全てのアドレス・インデックス・モードがダブル・ワード転送で可能です。アドレス・インデックス・モードの詳細については、3-21ページの第3.2.3.5節「アドレッシング・モード」を参照してください。

ダブル・ワード・モードでは、2番目の転送アドレスは常に最初の転送アドレスのLSBを反転したアドレスになります。つまり、最初の転送アドレスが偶数アドレスの場合は、2番目の転送は次に高いアドレスになります。一方、最初の転送アドレスが奇数アドレスの場合は、2番目の転送が次に低いアドレスとなります。

ダブル・ワード・モードは、DMSFCnのDBLWビットで設定します。リセット時は、DBLWがクリアされています(シングル・ワード・モードになります)。

## フレーム・カウント

フレーム・カウントはDMSFCnの8ビット・フィールドで、1つのブロック転送に含まれるフレーム数を指定します。フレーム・カウントの初期値は必要なフレーム数よりも1少ない値にします。たとえば、1つのブロックで8つのフレーム転送を行いたい場合は、フレーム・カウント・フィールドを7に設定します。希望のフレーム数が1の時は、フレーム・カウント・フィールドを0に設定します。リセット時、フレーム・カウント・フィールドは0(シングル・フレーム動作)に設定されています。なお、フレームの最大数は256です。

各フレーム転送完了ごとに、DMAコントローラによってフレーム・カウント・フィールドがデクリメントされます。自動初期化モードがイネーブルになっていれば、最終フレームが転送されると、DMAグローバル・フレーム・カウント・レジスタ(DMGFR)の値でフレーム・カウントがリロードされます。自動初期化機能の詳細については、3-27ページの第3.2.3.6節「自動初期化」を参照してください。

エレメント・カウントとフレーム・カウントを併用すれば、最大65536の転送が可能になります。転送総数は、エレメント・カウントとフレーム・カウントの積になります。

CPUとDMAコントローラが同時にフレーム・カウント・フィールドを変更しようとした場合は、CPUが優先されます。

### 3.2.3.4 転送モード制御レジスタ

DMA転送モード制御レジスタ(DMMCRn)は、チャンネルの転送モードを制御する16ビット・レジスタです。3-17ページの図3-4に、DMMCRnの構成を示します。レジスタ・フィールドの機能については、3-17ページの表3-10とそれ以降の節で説明します。

図3-4. DMA転送モード制御(DMMCRn)レジスタ

|          |      |      |       |      |      |     |      |      |     |   |   |   |   |
|----------|------|------|-------|------|------|-----|------|------|-----|---|---|---|---|
| 15       | 14   | 13   | 12    | 11   | 10   | 8   | 7    | 6    | 5   | 4 | 2 | 1 | 0 |
| AUTOINIT | DINM | IMOD | CTMOD | rsvd | SIND | DMS | rsvd | DIND | DMD |   |   |   |   |

表3-10. DMA転送モード制御(DMMCRn)レジスタのビット/フィールド説明

| ビット | 名前       | リセット値 | 機能                                                                       |
|-----|----------|-------|--------------------------------------------------------------------------|
| 15  | AUTOINIT | 0     | DMA自動初期化モード・ビット<br>AUTOINIT = 0 自動初期化がディセーブル<br>AUTOINIT = 1 自動初期化がイネーブル |

表3-10. DMA転送モード制御(DMMCRn)レジスタのビット/フィールド説明(続き)

| ビット    | 名前    | リセット値 | 機能                                                                                                                                                                                                                                                                                                                        |
|--------|-------|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 14     | DINM  | 0     | DMA割り込み発生マスク・ビット<br>DINM = 0 割り込み発生なし<br>DINM = 1 IMODビットに基づいて割り込みを発生                                                                                                                                                                                                                                                     |
| 13     | IMOD  | 0     | DMA割り込み発生モード・ビット<br>ABUモード(CTMOD=1)の場合<br>IMOD = 0 バッファが一杯になったときにのみ割り込みを発生<br>IMOD = 1 バッファの半分および全部が一杯になったときに割り込みを発生<br>マルチフレーム・モード(CTMOD=0)の場合<br>IMOD = 0 ブロック転送完了時に割り込みを発生<br>IMOD = 1 フレーム完了とブロック完了で割り込みを発生                                                                                                           |
| 12     | CTMOD | 0     | DMA転送カウンタ・モード制御ビット<br>CTMOD = 0 マルチフレーム・モード<br>CTMOD = 1 ABUモード                                                                                                                                                                                                                                                           |
| 11     | 予約    | 0     | 予約。このフィールドにライトされた値は無効です。                                                                                                                                                                                                                                                                                                  |
| 10 - 8 | SIND  | 0     | DMA転送元アドレス転送インデックス・モード・ビット<br>SIND = 000 修飾なし<br>SIND = 001 ポストインクリメント<br>SIND = 010 ポストデクリメント<br>SIND = 011 インデックスのオフセット(DMIDX0)でポストインクリメント<br>SIND = 100 インデックスのオフセット(DMIDX1)でポストインクリメント<br>SIND = 101 インデックスのオフセット(DMIDX0およびDMFRI0)でポストインクリメント<br>SIND = 110 インデックスのオフセット(DMIDX1およびDMFRI1)でポストインクリメント<br>SIND = 111 予約 |
| 7 - 6  | DMS   | 0     | DMA転送元アドレス空間選択ビット<br>DMS = 00 プログラム空間<br>DMS = 01 データ空間<br>DMS = 10 I/O空間<br>DMS = 11 予約                                                                                                                                                                                                                                  |
| 5      | 予約    | 0     | 予約。このフィールドにライトされた値は無効です。                                                                                                                                                                                                                                                                                                  |



表3-10. DMA転送モード制御(DMMCRn)レジスタのビット/フィールド説明(続き)

| ビット   | 名前   | リセット値 | 機能                                                                                                                                                                                                                                                                                                                        |
|-------|------|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4 - 2 | DIND | 0     | DMA転送先アドレス転送インデックス・モード・ビット<br>DIND = 000 修飾なし<br>DIND = 001 ポストインクリメント<br>DIND = 010 ポストデクリメント<br>DIND = 011 インデックスのオフセット(DMIDX0)でポストインクリメント<br>DIND = 100 インデックスのオフセット(DMIDX1)でポストインクリメント<br>DIND = 101 インデックスのオフセット(DMIDX0およびDMFRI0)でポストインクリメント<br>DIND = 110 インデックスのオフセット(DMIDX1およびDMFRI1)でポストインクリメント<br>DIND = 111 予約 |
| 1 - 0 | DMD  | 0     | DMA転送先アドレス空間選択ビット<br>DMD = 00 プログラム空間<br>DMD = 01 データ空間<br>DMD = 10 I/O空間<br>DMD = 11 予約                                                                                                                                                                                                                                  |

## 転送元アドレスと転送先アドレスの選択および修飾

3

DMMCR<sub>n</sub>のDMSとDMDのフィールドで、DMA転送に使われる転送元および転送先のアドレス空間を選択できます。DMSは転送元データがプログラム空間、データ空間、I/O空間のどこから転送されるのかを指定します。同様にDMDは転送先データがプログラム空間、データ空間、I/O空間のどこへ転送されるのかを指定します。これらの各モードのビット設定については、表3-10を参照してください。

DMMCR<sub>n</sub>ではさらに、転送元と転送先アドレス修飾も制御します。SINDフィールドでは、転送の進行においてどのように転送元アドレスを修飾するかを決定します。同様にDINDフィールドでは、転送の進行においてどのように転送先アドレスを修飾するかを決定します。アドレス修飾については、次のオプションを使用できます。

- ☐ 修飾なし(各転送に対してアドレスが一定のまま)
- ☐ 1ずつポストインクリメント
- ☐ 1ずつポストデクリメント
- ☐ エレメント・インデックス・レジスタ0(DMIDX0)にストアされているオフセット値でポストインクリメント
- ☐ エレメント・インデックス・レジスタ1(DMIDX1)にストアされているオフセット値でポストインクリメント
- ☐ エレメント・インデックス・レジスタ0(DMIDX0)とフレーム・インデックス・レジスタ0(DMFRI0)にストアされているオフセット値でポストインクリメント
- ☐ エレメント・インデックス・レジスタ1(DMIDX1)とフレーム・インデックス・レジスタ1(DMFRI1)にストアされているオフセット値でポストインクリメント

最後の4つのモードでは、エレメント・インデックス・レジスタとフレーム・インデックス・レジスタが使われています。エレメント・インデックス・レジスタ(DMIDX0とDMIDX1)は、転送中に転送元アドレスと転送先アドレスを修飾します。これらのレジスタにストアされているインデックスを使って、各エレメントの転送後に転送元アドレスまたは転送先アドレスが修飾されます。DMIDX0とDMIDX1は転送元と転送先のどちらのアドレスに対応するかは規定されていません。どちらのレジスタも、転送元か転送先、またはその両方を修飾するのに用いることができます。

フレーム・インデックス・レジスタ(DMFRI0とDMFRI1)は、エレメント転送のブロック完了(またはフレーム完了)後に転送元アドレスと転送先アドレスを修飾します。エレメントとフレームの両方でインデックスを用いると、各転送ごとにエレメント・インデックスによってアドレスが修飾され、各フレーム完了でフレーム・インデックスによってさらにアドレスが修飾されます。この機能を使用すれば、サーキュラ・バッファとソート機能を実行することができます(後述の関連節を参照)。DMFRI0とDMFRI1は、転送元と転送先のどちらのアドレスに対応するかは規定されていません。どちらのレジスタも、転送元か転送先、またはその両方を修飾するのに用いることができます。

DMMCR<sub>n</sub>は各チャンネルの設定が個別に行えるチャンネル・コンテキスト・レジスタです。なお、DMIDX0、DMIDX1、DMFRI0、DMFRI1はチャンネル・コンテキスト・レジスタではなく、DMAシステム全体のインデックス・オプションを設定するものです。

### 3.2.3.5 アドレッシング・モード

DMMCRnのCTMODフィールドは、各チャンネルのDMA転送カウンタの動作を制御します。マルチフレーム・モードでは、エレメント・インデックスとフレーム・インデックスを使って各転送後に転送元アドレスと転送先アドレスが修飾されます。転送するデータがフレームまたはブロックとしてフォーマットされている場合には、このモードが便利です。自動バッファリング機能を実行する場合には、ABUモードが使われます。このモードでは、各チャンネルのエレメント・カウンタ(DMCTRn)がバッファのサイズを表し、転送中にカウンタの修飾は行われません。転送元/転送先の例としてMcBSPが挙げられますが、使用する転送元/転送先は問いません。

### マルチフレーム・モード

マルチフレーム・モードでは、複数のエレメントが1つのフレームに収められ、複数のフレームが1つのブロックに収められた形でデータ転送が構成されます。エレメント・インデックス・レジスタ(DMIDX0とDMIDX1)はフレーム内の各エレメント・アドレスを修飾するのに使われ、フレーム・インデックス・レジスタ(DMFRI0とDMFRI1)はフレーム完了後のアドレス修飾に使われます。1フレームあたりの転送エレメント数はチャンネル・エレメント・カウント(DMCTRn)レジスタによって決まり、1ブロックあたりの転送フレーム数はチャンネル・フレーム・カウント(DMSFCnレジスタのフレーム・カウント・フィールド)によって決まります。エレメント・カウンタはエレメントが転送されるたび、フレーム・カウンタはフレームが完了するたびにデクリメントされます。

エレメント・カウントには16ビットの符号なし整数が使われます。エレメント・カウント・レジスタは、希望の転送エレメント数よりも1少ない値で初期化します。1フレームあたりの転送エレメント数は1(0000h)～65536(FFFFh)です。

フレーム・カウントには8ビットの符号なし整数が使われます。フレーム・カウント・レジスタは、希望の転送フレーム数よりも1少ない値で初期化します。1ブロックあたりの転送フレーム数は1(00h)～256(FFh)です。転送エレメントの総数をブロック・サイズといい、この数はフレーム・カウントとエレメント・カウントの積になります。

エレメント・カウンタは、転送ごとにデクリメントされます。フレーム内のエレメントがすべて転送されると、(シャドウ・レジスタにストアされていた)元の値でエレメント・カウンタがリロードされ、フレーム・カウンタがデクリメントされます。最終フレームの最終エレメントが転送されると、エレメント・カウンタとフレーム・カウンタの両方の値が0000hになり、自動初期化モードがイネーブル(AUTOINIT=1)でない限りこの値は変更されません。自動初期化モードの詳細については、3-27ページの第3.2.3.6節を参照してください。

「転送元アドレスと転送先アドレスの選択および修飾」節で述べたアドレス・インデックス・モードはすべてマルチフレーム・モードで有効です。なお、これらのインデックス・モードは、転送元アドレスと転送先アドレスを別々に設定することができます。次に、マルチフレーム動作の各アドレッシング・モードについて説明します。

- ☐ 修飾なし 転送後にアドレスが修飾されず、初期値が保持されます。
- ☐ ポストインクリメント 1回の転送が終わるごとにアドレス値が1ずつインクリメントされます。ダブル・ワード・モードが選択されている場合は、アドレス値は2ずつインクリメントされます。
- ☐ ポストデクリメント 1回の転送が終わるごとにアドレス値が1ずつデクリメントされます。ダブル・ワード・モードが選択されている場合は、アドレス値は2ずつデクリメントされます。
- ☐ インデックスのオフセット(DMIDX0またはDMIDX1)でポストインクリメント  
参照先のエレメント・インデックス・レジスタ値でアドレスが修飾されます。エレメント・インデックス・レジスタには、符号付き16ビット値(2の補数)がストアされています。転送後、次のアドレスを生成するために、このインデックス値によって現在のアドレスに加算(または減算)が行われます。ダブル・ワード・モードではインデックスの2倍の値でアドレスが修飾されます。
- ☐ インデックスのオフセット(DMIDX0とDMFRI0、またはDMIDX1とDMFRI1)またはソート・モードを用いてポストインクリメント  
現行の転送がフレーム内の最終エレメントでない場合は、参照先のエレメント・インデックス・レジスタの値でアドレスが修飾されます。現在のエレメントがフレーム内の最終エレメントである場合は、フレーム・インデックス・レジスタの値でアドレスが修飾されます。フレーム・インデックス・レジスタには、符号付き16ビット値がストアされています。このモードは、T1フレーム通信のようなデータのソートを行う場合に便利です。

### 例3-3. アドレス修飾によるデータのソート

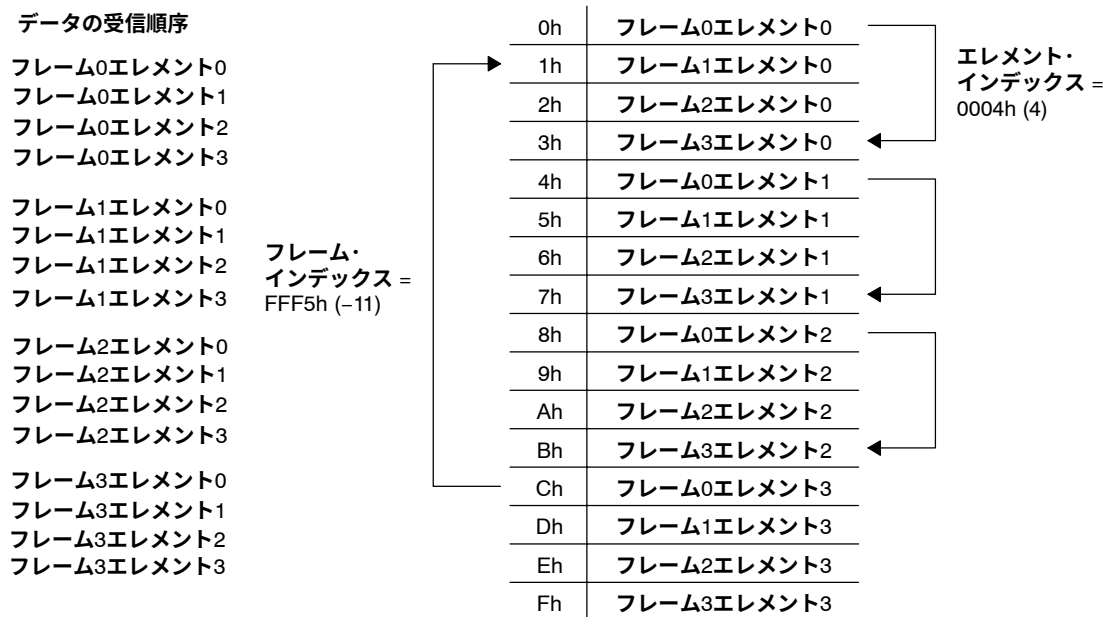
この例では、1フレームが4エレメント、1ブロックが4フレームで構成されたデータ列を仮定しています。転送元データは、図3-5の左側に示す順序で到着します。場合によっては、フレーム数ではなくエレメント数ごとにソートされたデータをストア/処理した方が都合がいいこともあります。その場合は、アドレス修飾でソート・モードを選択します。そしてこの例では、転送先エレメント・インデックス・レジスタを、各転送後に4インクリメントするように設定し、フレーム・インデックス・レジスタを各フレーム後に11デクリメントするように設定します。16回の転送で構成されるブロック全体の転送が完了すると、図3-5に示すようにフレーム番号ではなくエレメント番号でデータがソートされます。

エレメント・インデックスを4ずつインクリメントするように設定するには、エレメント・インデックス・レジスタに符号付き値0004hをストアします。また、フレーム・インデックスを11ずつデクリメントするように設定するには、フレーム・インデックス・レジスタに符号付き値0FFF5h(10進数の-11)をストアします。

ブロック内の最後のデータが転送された後、転送元アドレスと転送先アドレスは更新されず、どちらも最後のアドレスが保持されます。エレメント・カウンタとフレーム・カウンタの値は、最後のデータが転送されるとどちらも0000hになります。この例を実行するのに必要なDMAレジスタの設定については、3-56ページの例3-21「データのソートを伴うMcBSPからデータ・メモリーへの転送」を参照してください。

3

図3-5. データのソート例



## ABUモード

ABU(自動バッファリング)モードでは、DMA転送時の自動サーキュラ・バッファ機能を可能にしています。ABUモードのDMAコントローラとMcBSPを併用すると、バッファド・シリアル・ポート(BSP)上で実現される自動バッファリング機能と同じ機能が得られます。

このモードでは、転送元または転送先のどちらか一方が自動バッファリング・モードに設定され、もう一方はアドレス修飾なしとして設定されます。たとえば、McBSPの受信データをサーキュラ・バッファに転送する場合は、転送元をMcBSPのデータ受信レジスタ、転送先をDMAが制御するサーキュラ・バッファとなるように、DMAを設定する必要があります。

ABUモード動作中は、エレメント・カウント・レジスタにはバッファ・サイズがストアされています。フレーム・カウント・レジスタは、ABUモードでは機能しません。アドレスがバッファの最後に到達すると、自動的に先頭へ戻ります。このモードでは転送数が指定されないため、DMAチャンネルがディセーブルになるまでアドレスは変更され続けます。エレメント・カウント・レジスタは16ビットの符号なし整数として認識され、有効なバッファ・サイズ範囲は0002h～0FFFFhです。サイズはこの範囲内であればすべて有効で、2の累乗にする必要はありません。

このバッファには、ベース・アドレスと呼ばれる最小アドレスと、最大アドレスがあります。ベース・アドレスと最大アドレスの差がバッファ・サイズになります。バッファ・サイズは2の累乗でなくても構いませんが、ベース・アドレスは2の累乗にする必要があります。ベース・アドレスの位置はバッファ・サイズに依存します。ベース・アドレスは、バッファ・サイズの最上位ビット位置よりも、2の累乗数で1つ上(2進数で表す場合の1けた上)に相当するアドレス境界に配置していなければなりません。表3-11に、すべての有効バッファ・サイズに対するアドレス境界を示します。サーキュラ・バッファは、64Kのアドレス境界を超えることができません。この表では、考慮しない値はXで示しています。

表3-11. ABUバッファの例

| ABUのバッファ・サイズ<br>[16進数(10進数)] | バッファのベース・アドレス<br>[2進数] |
|------------------------------|------------------------|
| 0002h-0003h (2-3)            | XXXX XXXX XXXX XX00 b  |
| 0004h-0007h (4-7)            | XXXX XXXX XXXX X000 b  |
| 0008h-000Fh (8-15)           | XXXX XXXX XXXX 0000 b  |
| 0010h-001Fh (16-31)          | XXXX XXXX XXX0 0000 b  |
| 0020h-003Fh (32-63)          | XXXX XXXX XX00 0000 b  |
| 0040h-007Fh (64-127)         | XXXX XXXX X000 0000 b  |
| 0080h-00FFh (128-255)        | XXXX XXXX 0000 0000 b  |
| 0100h-01FFh (256-511)        | XXXX XXX0 0000 0000 b  |
| 0200h-03FFh (512-1023)       | XXXX XX00 0000 0000 b  |
| 0400h-07FFh (1024-2047)      | XXXX X000 0000 0000 b  |
| 0800h-0FFFh (2048-4095)      | XXXX 0000 0000 0000 b  |
| 1000h-1FFFh (4096-8191)      | XXX0 0000 0000 0000 b  |
| 2000h-3FFFh (8192-16383)     | XX00 0000 0000 0000 b  |
| 4000h-7FFFh (16384-32767)    | X000 0000 0000 0000 b  |
| 8000h-7FFFh (32768-65535)    | 0000 0000 0000 0000 b  |

## 例3-4. ABUバッファ・サイズの例

- バッファ・サイズが8(10進数)の場合、次に高い2の累乗数は16なので、バッファのベース・アドレスは16ワードの境界(0000h、0010h、0020hなど)に配置させる必要があります。
- バッファ・サイズが5(10進数)の場合、次に高い2の累乗数は8なので、バッファのベース・アドレスは8ワードの境界(0000h、0008h、0010h、0018h、0020hなど)に配置させる必要があります。
- バッファ・サイズが200(10進数)の場合、次に高い2の累乗数は256なので、バッファのベース・アドレスを256ワードの境界(0000h、0100h、0200h、0300hなど)に配置させる必要があります。

3

DMAは常に、チャンネルの転送元/転送先アドレス・レジスタで指定されたアドレスから転送を開始します。アドレス・レジスタは、バッファの指定範囲内であればどこでもポイントできます。各アクセスごとに、指定されたアドレス・インデックス・モードに従って適切なアドレス・レジスタが修飾されます。ABUモードに対しては、使用できるアドレス・インデックス・モードが限られています。表3-12に、ABUの動作で使用可能なモードを示します。

ABUモードを正しく設定するには、転送の一方(転送元あるいは転送先、両方は不可)をアドレス修飾なし(SIND/DIND=000b)に設定する必要があります。また、もう一方は修飾アドレス・インデックス・モードに設定する必要があります。表3-12に、ABUモードで有効なSINDとDINDの組み合わせを示します。これ以外の組み合わせを使用すると、予測不能な動作が起こる可能性があります。

表3-12. ABUのアドレス・インデックス・モード

| SIND | アドレス・インデックス・モード                     | DIND | アドレス・インデックス・モード                     |
|------|-------------------------------------|------|-------------------------------------|
| 000  | 修飾なし                                | 001  | ポストインクリメント                          |
|      |                                     | 010  | ポストデクリメント                           |
|      |                                     | 011  | インデックスのオフセット<br>(DMIDX0)でポストインクリメント |
|      |                                     | 100  | インデックスのオフセット<br>(DMIDX1)でポストインクリメント |
| 001  | ポストインクリメント                          | 000  | 修飾なし                                |
| 010  | ポストデクリメント                           |      |                                     |
| 011  | インデックスのオフセット<br>(DMIDX0)でポストインクリメント |      |                                     |
| 100  | インデックスのオフセット<br>(DMIDX1)でポストインクリメント |      |                                     |

転送の進行にしたがって、サーキュラ・バッファ内のアドレス値は1ずつインクリメント/デクリメントされるか、あるいはエレメント・インデックス・レジスタ(DMIDX0またはDMIDX1)の値で修飾されます。DMAチャンネルがダブル・ワード・モードに設定されている場合は、アドレスは2ずつインクリメント/デクリメントされるか、あるいはインデックス・オフセットの2倍の値で修飾されます。もし、計算された次のバッファ・アドレスがバッファの最大サイズを超えてしまう場合は、ベース・アドレスに戻ります。1よりも大きいインデックス値が使われる場合は、ベース・アドレスから対応するオフセットが計算されます。

ダブル・ワード・モードにおいて2番目の転送アドレスは、最大アドレスを超えているかどうかに関わらず、常に最初の転送アドレスのLSBを反転した値となります。ダブルワード転送の場合、バッファの折り返しアドレスは常に最初の転送アドレスに対して計算されます。

インデックス・アドレッシングでシングル・ワード転送を行う場合は、例3-5のようにバッファの折り返しアドレスが計算されます。インデックス・アドレッシングでダブル・ワード転送を行う場合は、例3-6のようにバッファの折り返しアドレスが計算されます。

#### 例3-5. インデックス・アドレッシングでシングル・ワード転送を行う場合の折り返しアドレス算出法

|         |             |
|---------|-------------|
| バッファ位置  | 0580h~059Eh |
| 現在のアドレス | 059Eh       |
| インデックス  | 2           |
| 次のアドレス  | 0581h       |

#### 例3-6. インデックス・アドレッシングでダブル・ワード転送を行う場合の折り返しアドレス算出法

|         |                         |
|---------|-------------------------|
| バッファ位置  | 0580h~059Eh             |
| 現在のアドレス | 059Eh                   |
| インデックス  | 2(ダブル・ワード・モードのため合計数は+4) |
| 次のアドレス  | 0583h                   |

また、負の値をインデックスとして指定することもできます。この場合、ベース・アドレスと最大アドレスの算出法は上の例と同じになります。アドレスは負のアドレス・インデックスによって修飾され、ベース・アドレスを超えたらアドレスが最大アドレスに戻ります。-1よりも小さいインデックスが使われると、最大アドレスからの対応するオフセット分が計算されます。

ABUモードでは、割り込みをバッファの半分または全部が一杯になったときに発生するように設定できます。割り込み処理の詳細については、3-27ページの第3.2.3.7節「割り込み生成」を参照してください。



### 3.2.3.6 自動初期化

DMMCRnのAUTOINITフィールドは、各DMAチャンネルの自動初期化機能を制御します。自動初期化機能がイネーブル(AUTOINIT=1)になっている場合は、ブロック転送の完了時にチャンネル・コンテキスト・レジスタが再初期化されます。自動初期化モードを使用すれば、ブロック転送の完了時にCPUが介入する必要がなくなります。自動初期化が行われると、次のチャンネル・コンテキスト・レジスタが修飾されます。

- ☐ DMSRCnには、DMGSA(グローバル転送元アドレス・レジスタ)の値がロードされます。DMGSAには16ビットの転送元アドレスがストアされています。
- ☐ DMDSTnには、DMGDA(グローバル転送先アドレス・レジスタ)の値がロードされます。DMGDAには16ビットの転送先アドレスがストアされています。
- ☐ DMCTRnには、DMGCR(グローバル・エレメント・カウント・レジスタ)の値がロードされます。DMGCRには、16ビットの符号なしエレメント・カウント値がストアされています。
- ☐ DMSFCnのフレーム・カウント・フィールドには、DMGFR(グローバル・フレーム・カウント・レジスタ)の値がロードされます。DMGFRには8ビットの符号なしフレーム・カウント値がストアされています。DMGFRの上位8ビットは予約領域で無効です。予約されたビットは常に0としてロードされます。

自動初期化はマルチフレーム・モード(CTMOD=0)でのみ使用できます。

### 3.2.3.7 割り込み発生

DMMCRnの2つのフィールド(DINMとIMOD)は、DMA転送時の割り込み処理を制御します。DINMは、転送の一部または全部の完了に伴う割り込み発生をイネーブル/ディセーブルにします。DINM=0の場合は割り込みがディセーブルになり、割り込みが発生しません。DINM=1の場合は、割り込みがイネーブルになり、IMODビットの設定に基づいて割り込みが発生します。CPUの割り込みマスク・レジスタ(IMR)とINTMビットによって、DMAからの割り込みを処理するかどうか制御します。そして、DINMはDMAによる割り込みを可能にするものです。IMR、INITM、および割り込み処理の詳細については、ユーザーズ・ガイド『TMS320C54x DSP, CPU and Peripherals, Reference Set, Volume 1』(文献番号SPRU131)を参照してください。

割り込み発生がイネーブル(DINM=1)の時は、DMMCRnのIMODフィールドで、ブロック転送中の割り込みを発生させるタイミングを制御します。表3-13に、使用可能な動作モードを示します。

表3-13. DMAブロック転送の割り込み発生モード

| モード         | CTMOD | DINM | IMOD | 割り込み発生                |
|-------------|-------|------|------|-----------------------|
| ABU         | 1     | 1    | 0    | バッファが一杯になったときのみ       |
| ABU         | 1     | 1    | 1    | バッファの半分および全部が一杯になったとき |
| マルチフレーム     | 0     | 1    | 0    | ブロック転送完了時             |
| マルチフレーム     | 0     | 1    | 1    | フレーム完了時とブロック完了時       |
| ABU/マルチフレーム | X     | 0    | X    | 割り込み発生なし              |

マルチフレーム・モードでは、各フレーム転送完了時(およびフレームで構成される各ブロックの転送完了時)か、またはブロック全体の転送完了時のみ割り込みを発生させることができます。フレーム・カウントが0になったときがフレームの終端であり、エレメント・カウントとフレーム・カウントの両方が0になったときがブロックの転送完了です。

ABUモード(CTMOD=1)では、バッファ全体のデータが転送されたときか、バッファの半分のデータが転送されるごとに割り込みを発生させることができます。このハーフ・バッファ・オプションは、'C548や'C549などのバッファド・シリアル・ポート(BSP)と同じような動作をします。

割り込みを発生させるタイミングは、現行アドレスの次のアドレスを参照して決定します。バッファの中間点の決め方は、使用する'C54x DSPによって異なります。各デバイスのハーフ・バッファ割り込み発生については、以降の節で詳しく説明します。

### 3.2.3.8 'C5402/'C5420におけるABUモードでのハーフ・バッファ割り込み発生

'C5402と'C5420では、次のアドレスがバッファの中間点以上になったとき(正のインデックス使用時)か、中間点を下回ったとき(負のインデックス使用時)にハーフ・バッファ割り込みが発生します。割り込みのポイントは、バッファ・サイズが奇数か偶数かによって異なります。フル・バッファ割り込みについては、次のアドレスがベース・アドレスに戻るとき(正のインデックス)か、最大アドレスに戻るとき(負のインデックス)に発生します。以降の例で、それぞれの場合に割り込みが発生するタイミングを説明します。

## 例3-7. 'C5402/'C5420におけるABU割り込み例 +1インデックス使用時でバッファ・サイズが偶数の場合

バッファ・サイズ=8

ハーフ・バッファ=4

| 現行アドレス | 次のアドレス | 割り込み         |
|--------|--------|--------------|
| 0000h  | 0001h  |              |
| 0001h  | 0002h  |              |
| 0002h  | 0003h  |              |
| 0003h  | 0004h  | 割り込み発生       |
| 0004h  | 0005h  |              |
| 0005h  | 0006h  |              |
| 0006h  | 0007h  |              |
| 0007h  | 0000h  | 割り込み発生(折り返し) |

3

## 例3-8. 'C5402/'C5420におけるABU割り込み例 +1インデックス使用時でバッファ・サイズが奇数の場合

バッファ・サイズ=7

ハーフ・バッファ=3

| 現行アドレス | 次のアドレス | 割り込み         |
|--------|--------|--------------|
| 0000h  | 0001h  |              |
| 0001h  | 0002h  |              |
| 0002h  | 0003h  | 割り込み発生       |
| 0003h  | 0004h  |              |
| 0004h  | 0005h  |              |
| 0005h  | 0006h  |              |
| 0006h  | 0000h  | 割り込み発生(折り返し) |

## 例3-9. 'C5402/'C5420におけるABU割り込み例 -1インデックス使用時でバッファ・サイズが偶数の場合

バッファ・サイズ=8

ハーフ・バッファ=4

| 現行アドレス | 次のアドレス | 割り込み         |
|--------|--------|--------------|
| 0007h  | 0006h  |              |
| 0006h  | 0005h  |              |
| 0005h  | 0004h  |              |
| 0004h  | 0003h  | 割り込み発生       |
| 0003h  | 0002h  |              |
| 0002h  | 0001h  |              |
| 0001h  | 0000h  |              |
| 0000h  | 0007h  | 割り込み発生(折り返し) |

## 例3-10. 'C5402/'C5420におけるABU割り込み例 -1インデックス使用時でバッファ・サイズが奇数の場合

バッファ・サイズ=7

ハーフ・バッファ=3

| 現行アドレス | 次のアドレス | 割り込み         |
|--------|--------|--------------|
| 0006h  | 0005h  |              |
| 0005h  | 0004h  |              |
| 0004h  | 0003h  |              |
| 0003h  | 0002h  | 割り込み発生       |
| 0002h  | 0001h  |              |
| 0001h  | 0000h  |              |
| 0000h  | 0006h  | 割り込み発生(折り返し) |

### 3.2.3.9 'C5410におけるABUモードでのハーフ・バッファ割り込み発生

'C5410では、次のアドレスがバッファの中間点を越えたとき(正のインデックス使用時)か、中間点を下回ったとき(負のインデックス使用時)にハーフ・バッファ割り込みが発生します。割り込みのポイントは、バッファ・サイズが奇数か偶数かによって異なります。フル・バッファ割り込みは、次のアドレスがベース・アドレスに戻るとき(正のインデックス)か、最大アドレスに戻るとき(負のインデックス)に発生します。以降の例で、それぞれの場合に割り込みが発生するタイミングを説明します。

#### 例3-11. 'C5410におけるABU割り込み例 +1インデックス使用時でバッファ・サイズが偶数の場合

バッファ・サイズ=8

ハーフ・バッファ=4

| 現行アドレス | 次のアドレス | 割り込み         |
|--------|--------|--------------|
| 0000h  | 0001h  |              |
| 0001h  | 0002h  |              |
| 0002h  | 0003h  |              |
| 0003h  | 0004h  |              |
| 0004h  | 0005h  | 割り込み発生       |
| 0005h  | 0006h  |              |
| 0006h  | 0007h  |              |
| 0007h  | 0000h  | 割り込み発生(折り返し) |

#### 例3-12. 'C5410におけるABU割り込み例 +1インデックス使用時でバッファ・サイズが奇数の場合

バッファ・サイズ=7

ハーフ・バッファ=3

| 現行アドレス | 次のアドレス | 割り込み         |
|--------|--------|--------------|
| 0000h  | 0001h  |              |
| 0001h  | 0002h  |              |
| 0002h  | 0003h  |              |
| 0003h  | 0004h  | 割り込み発生       |
| 0004h  | 0005h  |              |
| 0005h  | 0006h  |              |
| 0006h  | 0000h  | 割り込み発生(折り返し) |

例3-13. 'C5410におけるABU割り込み例 -1インデックス使用時でバッファ・サイズが偶数の場合

| バッファ・サイズ=8 |        | ハーフ・バッファ=4   |
|------------|--------|--------------|
| 現行アドレス     | 次のアドレス | 割り込み         |
| 0007h      | 0006h  |              |
| 0006h      | 0005h  |              |
| 0005h      | 0004h  |              |
| 0004h      | 0003h  | 割り込み発生       |
| 0003h      | 0002h  |              |
| 0002h      | 0001h  |              |
| 0001h      | 0000h  |              |
| 0000h      | 0007h  | 割り込み発生(折り返し) |

例3-14. 'C5410におけるABU割り込み例 -1インデックス使用時でバッファ・サイズが奇数の場合

| バッファ・サイズ=7 |        | ハーフ・バッファ=3   |
|------------|--------|--------------|
| 現行アドレス     | 次のアドレス | 割り込み         |
| 0006h      | 0005h  |              |
| 0005h      | 0004h  |              |
| 0004h      | 0003h  |              |
| 0003h      | 0002h  | 割り込み発生       |
| 0002h      | 0001h  |              |
| 0001h      | 0000h  |              |
| 0000h      | 0006h  | 割り込み発生(折り返し) |

3.3 拡張アドレッシング

DMAコントローラは、拡張プログラム・メモリー空間との間でデータ転送を行うことができます。この機能を実現するために、2つのサブアドレス・レジスタ、DMA転送元プログラム・ページ・アドレス・レジスタ(DMSRCP)とDMA転送先プログラム・ページ・アドレス・レジスタ(DMDSTP)が提供されています。これらのレジスタには、それぞれ転送元と転送先を示す拡張プログラム・ページ数がストアされています。図3-6と図3-7に示すように、これら各レジスタの下位7ビットに、転送元と転送先の拡張アドレス・ページをストアします。リセット時は、DMSRCPとDMDSTPが0000h(プログラム・ページ0)に初期化されます。予約されたビットは常に0としてリードされます。

DMSRCPとDMDSTPは各チャンネルごとの設定ではなく、すべてのDMAチャンネルに対するプログラム・メモリーの転送元/転送先のアドレス・ページを指定します。また、転送元/転送先のアドレスを修飾中に、これらのレジスタに格納されたプログラム・ページ・アドレスは変更されません。したがって、プログラム空間での転送は64Kのページ境界を超えることができません。転送中にプログラムのページ境界を超えてしまうと、次の転送は同じページに戻ります。

すべての'C54xデバイスが128ページの拡張プログラム・メモリーをサポートしているわけではありません。拡張メモリーの詳細については、各デバイスのデータ・シートを参照してください。

図3-6. DMA転送元プログラム・ページ・アドレス・レジスタ(DMSRCP)

|    |   |                   |   |
|----|---|-------------------|---|
| 15 | 7 | 6                 | 0 |
| 予約 |   | 転送元プログラム・ページ・アドレス |   |

図3-7. DMA転送先プログラム・ページ・アドレス・レジスタ(DMDSTP)

|    |   |                   |   |
|----|---|-------------------|---|
| 15 | 7 | 6                 | 0 |
| 予約 |   | 転送先プログラム・ページ・アドレス |   |

### 3.4 DMAメモリー・マップ

この節では、'C5402、'C5410、'C5420のDMAメモリー・マップについて説明します。

#### 3.4.1 'C5402のDMAメモリー・マップ

'C5402のDMAでは、内部アクセスのみサポートしています。表3-14に'C5402のDMAメモリー・マップを示します。「予約」と記されたメモリー領域(網掛け部分)には、'C5402のDMAからアクセスすることはできません。このDMAメモリー・マップは、MP/MC、DROM、またはOVLYビットの影響を受けません。

表3-14. 'C5402のDMAメモリー・マップ

| アドレス範囲(16進数) |                 | 説明                     |
|--------------|-----------------|------------------------|
| プログラム空間      | 00 0000 0F FFFF | 予約                     |
| データ空間        | 0000 001F       | 予約                     |
|              | 0020            | McBSP0データ受信レジスタ(DRR20) |
|              | 0021            | McBSP0データ受信レジスタ(DRR10) |
|              | 0022            | McBSP0データ送信レジスタ(DXR20) |
|              | 0023            | McBSP0データ送信レジスタ(DXR10) |
|              | 0024 003F       | 予約                     |
|              | 0040            | McBSP1データ受信レジスタ(DRR21) |
|              | 0041            | McBSP1データ受信レジスタ(DRR11) |
|              | 0042            | McBSP1データ送信レジスタ(DXR21) |
|              | 0043            | McBSP1データ送信レジスタ(DXR11) |
|              | 0044 005F       | 予約                     |
|              | 0060 007F       | スクラッチパッドRAM            |
|              | 0080 3FFF       | DARAM                  |
|              | 4000 FFFF       | 予約                     |
| I/O空間        | 0000 FFFF       | 予約                     |



## 3.4.2 'C5410のDMAメモリー・マップ

'C5410では、内部メモリーと外部メモリーへのDMAアクセスをサポートしています。表3-15に、'C5410のDMAメモリー・マップを示します。「予約」と記されたメモリー領域(網掛け部分)には、'C5410のDMAからアクセスすることはできません。このDMAメモリー・マップは、MP/ $\overline{MC}$ 、DROM、またはOVLYビットの影響を受けません。

3

表3-15. 'C5410のDMAメモリー・マップ

|         | アドレス範囲(16進数) |         | 説明                     |
|---------|--------------|---------|------------------------|
| プログラム空間 | 00 0000      | 00 007F | 予約                     |
|         | 00 0080      | 00 BFFF | 外部メモリー                 |
|         | 00 C000      | 00 FFFF | オンチップROM               |
|         | 10 0000      | 01 7FFF | 外部メモリー                 |
|         | 01 8000      | 01 FFFF | オンチップSARAM2            |
|         | 02 0000      | 7F FFFF | 外部メモリー                 |
| データ空間   | 0000         | 001F    | 予約                     |
|         | 0020         |         | McBSP0データ受信レジスタ(DRR20) |
|         | 0021         |         | McBSP0データ受信レジスタ(DRR10) |
|         | 0022         |         | McBSP0データ送信レジスタ(DXR20) |
|         | 0023         |         | McBSP0データ送信レジスタ(DXR10) |
|         | 0024         | 002F    | 予約                     |
|         | 0030         |         | McBSP2データ受信レジスタ(DRR22) |
|         | 0031         |         | McBSP2データ受信レジスタ(DRR12) |
|         | 0032         |         | McBSP2データ送信レジスタ(DXR22) |
|         | 0033         |         | McBSP2データ送信レジスタ(DXR12) |
|         | 0034         | 0035    | 予約                     |
|         | 0036         |         | RCERA2                 |
|         | 0037         |         | XCERA2                 |
|         | 0038         | 0039    | 予約                     |
|         | 003A         |         | RCERA0                 |
|         | 003B         |         | XCERA0                 |

## DMAメモリー・マップ

表3-15. 'C5410のDMAメモリー・マップ (続き)

| アドレス範囲(16進数) |      | 説明                     |
|--------------|------|------------------------|
| 003C         | 003F | 予約                     |
| 0040         |      | McBSP1データ受信レジスタ(DRR21) |
| 0041         |      | McBSP1データ受信レジスタ(DRR11) |
| 0042         |      | McBSP1データ送信レジスタ(DXR21) |
| 0043         |      | McBSP1データ送信レジスタ(DXR11) |
| 0044         | 0049 | 予約                     |
| 004A         |      | RCERA1                 |
| 004B         |      | XCERA1                 |
| 004C         | 005F | 予約                     |
| 0060         | 7FFF | オンチップRAM               |
| 8000         | FFFF | 外部メモリー                 |
| I/O空間        | 0000 | FFFF 外部メモリー            |

### 3.4.3 'C5420のDMAメモリー・マップ

'C5420のDMAでは、内部アクセスのみサポートしています。表3-16に、'C5420のDMAメモリー・マップを示します。「予約」と記されたメモリー領域(網掛け部分)には、'C5420のDMAからアクセスすることはできません。なお、どちらのサブシステムでも、I/O空間全体が内部プロセッサ間FIFO専用となります。したがって、'C5420のDMAが外部のI/Oポートにアクセスすることはできません。このDMAメモリー・マップは、MP/ $\overline{MC}$ 、DROM、またはOVLYビットの影響を受けません。

表3-16. 'C5420のDMAメモリー・マップ

|         | アドレス範囲(16進数) |         | 説明                     |
|---------|--------------|---------|------------------------|
| プログラム空間 | 00 0000      | 00 001F | 予約                     |
|         | 00 0020      |         | McBSP0データ受信レジスタ(DRR20) |
|         | 00 0021      |         | McBSP0データ受信レジスタ(DRR10) |
|         | 00 0022      |         | McBSP0データ送信レジスタ(DXR20) |
|         | 00 0023      |         | McBSP0データ送信レジスタ(DXR10) |
|         | 00 0024      | 00 002F | 予約                     |
|         | 00 0030      |         | McBSP2データ受信レジスタ(DRR22) |
|         | 00 0031      |         | McBSP2データ受信レジスタ(DRR12) |
|         | 00 0032      |         | McBSP2データ送信レジスタ(DXR22) |
|         | 00 0033      |         | McBSP2データ送信レジスタ(DXR12) |
|         | 00 0034      | 00 003F | 予約                     |
|         | 00 0040      |         | McBSP1データ受信レジスタ(DRR21) |
|         | 00 0041      |         | McBSP1データ受信レジスタ(DRR11) |
|         | 00 0042      |         | McBSP1データ送信レジスタ(DXR21) |
|         | 00 0043      |         | McBSP1データ送信レジスタ(DXR11) |
|         | 00 0044      | 00 005F | 予約                     |
|         | 00 0060      | 00 3FFF | オンチップDARAM0            |
|         | 00 4000      | 00 7FFF | オンチップSARAM1            |
|         | 00 8000      | 00 FFFF | オンチップSARAM2            |
|         | 01 0000      | 01 005F | 予約                     |
|         | 01 0060      | 01 3FFF | オンチップDARAM0            |
|         | 01 4000      | 01 7FFF | オンチップSARAM1            |
|         | 01 8000      | 01 FFFF | オンチップSARAM3            |
|         | 02 0000      | 02 005F | 予約                     |
|         | 02 0060      | 02 3FFF | オンチップDARAM0            |
|         | 02 4000      | 02 7FFF | オンチップSARAM1            |
|         | 02 8000      | 02 EFFF | 予約                     |
|         | 02 F000      | 02 FFFF | オンチップSARAM4            |

## DMAメモリー・マップ

表3-16. 'C5420のDMAメモリー・マップ (続き)

|       | アドレス範囲(16進数) |         | 説明                     |
|-------|--------------|---------|------------------------|
|       | 03 0000      | 03 005F | 予約                     |
|       | 03 0060      | 03 3FFF | オンチップDARAM0            |
|       | 03 4000      | 03 7FFF | オンチップSARAM1            |
|       | 03 8000      | 03 FFFF | 予約                     |
| データ空間 | 0000         | 001F    | 予約                     |
|       |              |         | McBSP0データ受信レジスタ(DRR20) |
|       | 0021         |         | McBSP0データ受信レジスタ(DRR10) |
|       | 0022         |         | McBSP0データ送信レジスタ(DXR20) |
|       | 0023         |         | McBSP0データ送信レジスタ(DXR10) |
|       | 0024         | 002F    | 予約                     |
|       | 0030         |         | McBSP2データ受信レジスタ(DRR22) |
|       | 0031         |         | McBSP2データ受信レジスタ(DRR12) |
|       | 0032         |         | McBSP2データ送信レジスタ(DXR22) |
|       | 0033         |         | McBSP2データ送信レジスタ(DXR12) |
|       | 0034         | 003F    | 予約                     |
|       | 0040         |         | McBSP1データ受信レジスタ(DRR21) |
|       | 0041         |         | McBSP1データ受信レジスタ(DRR11) |
|       | 0042         |         | McBSP1データ送信レジスタ(DXR21) |
|       | 0043         |         | McBSP1データ送信レジスタ(DXR11) |
|       | 0044         | 005F    | 予約                     |
|       | 0060         | 007F    | オンチップ・スクラッチパッドRAM      |
|       | 0080         | 3FFF    | オンチップDARAM0            |
|       | 4000         | 7FFF    | オンチップSARAM1            |
|       | 8000         | FFFF    | オンチップSARAM2            |
| I/O空間 | 0000         | FFFF    | 内部プロセッサ間通信用のDMA FIFO   |

### 3.5 DMA転送のレイテンシー

16ビットのDMA転送はすべて、リードの後にライトが行われます。これらの動作が完了する時間は、転送元/転送先の位置、外部インターフェイス条件(ウェイトステートやバンク・スイッチング・サイクルなど)、動作中のDMAチャネルの数、ホスト・ポート・インターフェイス(HPIx)の動作レベルによって異なってきます。

内部(オンチップ)の転送元から内部の転送先にDMA転送を1回行うには、4 CPUクロック・サイクルかかります。この場合、転送のリード部分で2サイクル、ライト部分で2サイクルかかります。リードかライト、あるいはその両方で外部(オフチップ)にアクセスすると、総転送時間はその通信アクセス完了までに要するサイクル数だけ長くなります。この時間によって、所定の転送元/転送先間の転送の最大転送速度が決まります。’C54xデバイスはすべて、内部間の転送を4サイクルで行います。したがって、1チャネルの最大転送速度は、CPUクロック速度の1/4になります。たとえば、CPUクロック速度が100MHzで動作するプロセッサであれば、1秒あたり最大で25Mの16ビット転送(25Mワード/秒)か、12.5Mのダブル・ワード(32ビット)転送を行います。表3-17に、DMAの最大転送サイクル・タイムの一覧を示します。オンチップ・メモリーまたはメモリー・マップド・レジスタへのアクセスは、どちらも内部アクセスと見なします。DMAによる外部アクセスをサポートしていないデバイスについては、表内の項目に「なし」と記します。

表3-17. DMA転送サイクル・タイム

| DMA転送元位置 | DMA転送先位置 | ’C5402の転送<br>サイクル | ’C5410の転送<br>サイクル | ’C5420の転送<br>サイクル |
|----------|----------|-------------------|-------------------|-------------------|
| 内部       | 内部       | 4                 | 4                 | 4                 |
| 内部       | 外部       | なし                | 5                 | なし                |
| 外部       | 内部       | なし                | 5                 | なし                |
| 外部       | 外部       | なし                | 6                 | なし                |

外部アクセスを必要とする転送では、転送完了までにかかるサイクル数は、外部インターフェイス条件(ソフトウェアおよびハードウェアのウェイト・ステート、バンク・スイッチングによる追加サイクル、CLKOUT分周モードでの通信速度など)に影響を受けます。たとえば、’C5410では、CLKOUT(および外部パラレル・インターフェイス)をCPUクロック速度の1/4に設定することができます。この場合は、設定に従って外部のDMA転送の速度も低下します。外部アクセスをとまなうDMAの最大転送速度を概算するときには、実際のアプリケーションにおける諸条件を考慮してください。

表3-17の'C5410の欄では、外部インターフェイスのハードウェア/ソフトウェアによるウェイト・ステートはなく、CLKOUTが1分周モードになっている(CLKOUTとCPUクロックが同じ)ものと仮定しています。

あるチャンネルの転送速度は、他のチャンネルの動作にも影響を受けます。高優先順位のチャンネルは順次先に処理されるため、あるチャンネルのデータ速度は、同じ優先順位でアクティブになっている他のチャンネル数によって決まります。したがって、高優先順位のチャンネルがすべてアクティブである間、高優先順位の各チャンネルの最大転送速度は、高優先順位のチャンネル総数分の1になります。一方、低優先順位のチャンネルの転送速度は、高優先順位の処理が終了するまでは0のままです。例3-15で、このような状況について説明します。

#### 例3-15. DMAチャンネル転送速度の例

この例では、'C54xプロセッサのCPUクロックが100MHzで動作していることを前提とします。なお、DMAチャンネルは、次のように設定されているものとします。

- ☐ DMAチャンネル1 100個のシングル・ワード転送、内部メモリー間、高優先順位
- ☐ DMAチャンネル2 200個のシングル・ワード転送、内部メモリー間、高優先順位
- ☐ DMAチャンネル3 50個のシングル・ワード転送、内部メモリー間、低優先順位
- ☐ DMAチャンネル4 75個のシングル・ワード転送、内部メモリー間、低優先順位
- ☐ DMAチャンネル5、6はディセーブル。HPI-8/-16はインアクティブ

ここでは、チャンネル1、2、3、4が同時にイネーブルになり、どのチャンネルも同期イベントを待たないこと、チャンネル1と2が先に処理され、チャンネル3と4はチャンネル1と2が完了するまで待機することを前提とします。各チャンネルの転送速度は、次のようになります。

チャンネル1と2がアクティブな時

- ☐ DMAチャンネル1のデータ転送速度=25MWps/アクティブ・チャンネル数2=12.5MWps
- ☐ DMAチャンネル2のデータ転送速度=25MWps/アクティブ・チャンネル数2=12.5MWps
- ☐ DMAチャンネル3のデータ転送速度=優先順位により0
- ☐ DMAチャンネル4のデータ転送速度=優先順位により0

チャンネル1は完了したが、チャンネル2がまだアクティブな時

- ☐ DMAチャンネル1のデータ転送速度=0(完了)
- ☐ DMAチャンネル2のデータ転送速度=25MWps/アクティブ・チャンネル数1=25MWps
- ☐ DMAチャンネル3のデータ転送速度=優先順位により0
- ☐ DMAチャンネル4のデータ転送速度=優先順位により0

チャンネル1と2が完了し、チャンネル3と4がアクティブになった時

- ☐ DMAチャンネル1のデータ転送速度=0(完了)
- ☐ DMAチャンネル2のデータ転送速度=0(完了)
- ☐ DMAチャンネル3のデータ転送速度=25MWps/アクティブ・チャンネル数2=12.5MWps
- ☐ DMAチャンネル4のデータ転送速度=25MWps/アクティブ・チャンネル数2=12.5MWps

チャンネル1、2、3は完了したが、チャンネル4がまだアクティブな時

- ☐ DMAチャンネル1のデータ転送速度=0(完了)
- ☐ DMAチャンネル2のデータ転送速度=0(完了)
- ☐ DMAチャンネル3のデータ転送速度=0(完了)
- ☐ DMAチャンネル4のデータ転送速度=25MWps/アクティブ・チャンネル数1=25MWps

### 3.6 DMAコントローラを介した拡張ホスト・ポート・インターフェイスのアクセス

3

拡張ホスト・ポート・インターフェイス(HPI8/HPI16)は、DMAバスを使ってオンチップ・メモリーにアクセスします。HPIはDMAコントローラ上に専用ポートを持ち、DMAバスの使用要求を出します。DMAコントローラは、現行の動作に基づいてこれらの要求を処理します。HPIxが要求を出すと、現行のDMA転送が完了してから、DMAコントローラがHPIxにDMAバスへのアクセス許可を与えます。優先度の高低に関わらず、未処理の転送はすべてHPIxがバスを解放するまで中断されます。HPIがバスを解放すると、未処理のDMA転送がそれまで通りに続行されます。したがって、HPIxの動作はDMAチャンネルの転送速度に影響を与えません。

ダブル・ワード転送では、HPIx要求による割り込みが発生しません。この場合は、両方の16ビット転送が完了してから、DMAコントローラがHPIxにアクセス許可を与えます。

HPIxはDMAコントローラ上に専用ポートを持ちますが、専用チャンネルはありません。HPIxがDMAバスを使用する際、各チャンネルのコンテキスト処理は必要ありません。通常、HPIxはホストによって制御されますが、内部バスへのアクセスはDMAコントローラによって処理されます。よって、HPIxのためにDMAチャンネルが使われることはありません。

拡張ホスト・ポート・インターフェイス(HPI8/HPI16)の動作方法の詳細については、第4章「拡張8ビット・ホスト・ポート・インターフェイス(HPI-8)」と第5章「拡張16ビット・ホスト・ポート・インターフェイス(HPI-16)」を参照してください。



### 3.7 'C5420の内部でのプロセッサ間FIFOによる通信

'C5420には、互いに独立した2つのCPUサブシステムがあり、それぞれ全機能が独立して動作します。

サブシステム間のデータ交換方法の1つに、DMAコントローラとオンチップFIFO(First-in, First-out)ユニットを併用する方法があります。FIFOはDMAのI/O空間にマッピングされています。'C5420でDMAのI/O空間にアクセスする場合は、必ずこのFIFOへリードまたはライトを行います。すべてのアドレスは、同じFIFO位置にマッピングされています。このため、'C5420のDMAでは、他のC54x製品と同様のI/O空間へアクセスすることができません。

'C5420サブシステムの通信方法の詳細については、第6章「内部でのプロセッサ間通信」を参照してください。

3

### 3.8 パワーダウン・モードでのDMA動作

'C54xデバイスには、いくつかのパワーダウン・モードが用意されています。これらのモードを使用すれば、デバイスの一部または全部を休止状態にして、通常動作時よりも消費電力を節約することができます。パワーダウン・モードは、IDLE命令を実行するか、HMステータス・ビットを1にセットした上でHOLD入力をローにドライブすることで実現できます。また、他のペリフェラルと同様にDMA割り込みを使ってCPUをIDLE状態から復帰させることができます。

DMAは、どのIDLEモード(IDLE1/2/3)にいても動作を続けます。DMAが動作していない(転送を実行していない)ときには、DMAコントローラが自動的に内部クロックを停止して消費電力を節約します。転送の実行が必要になると、クロックがオンになり、転送が完了すると再度オフに切り替わります。このクロック管理はどのIDLEモードでも行われます。

IDLE1とIDLE2では、DMAのクロック・ソースは、CPUクロックとシステム・クロックの生成に使われているものと同じソース(PLLまたはX2/CLKINの分周)になります。IDLE3ではPLLが停止されるため、PLLをDMAのクロック・ソースとして使うことはできません。この場合は、DMAコントローラのクロックとしてX2/CLKINが直接使われます。このため、IDLE3モードでDMAが動作するためには常にX2/CLKINピンにクロックが供給されていなければなりません。

IDLE3モードでDMAのクロックとしてX2/CLKINが使われる場合は、分周も逡倍も行われません。したがって、IDLE3モードではDMA転送速度にも影響が出ます。

### 3.9 プログラム例

以降に、DMAコントローラのプログラム例をいくつか示します。各例には、各動作の説明と、DMAコントローラ設定用プログラムが記載されています。ここに記載されているのは設定プログラムのみで、割り込みサービス・ルーチンなどは含まれていません。各プログラムでは、分かりやすいようにDMAレジスタの名前が参照されています。次ページに示すように、これらのレジスタのアドレスはアセンブリ・コードで定義することができます。

```
*****
*
* 54x Register Definitions for the DMA Controller
*
*****
DMPREC .set 0054h ;Channel Priority and Enable Control Register
DMSA .set 0055h ;Sub-bank Address Register
DMSDI .set 0056h ;Sub-bank Data Register with autoincrement
DMSDN .set 0057h ;Sub-bank Data Register without modification
DMSRC0 .set 00h ;Channel 0 Source Address Register
DMDST0 .set 01h ;Channel 0 Destination Address Register
DMCTR0 .set 02h ;Channel 0 Element Count Register
DMSFC0 .set 03h ;Channel 0 Sync Select and Frame Count Register
DMMCR0 .set 04h ;Channel 0 Transfer Mode Control Register
DMSRC1 .set 05h ;Channel 1 Source Address Register
DMDST1 .set 06h ;Channel 1 Destination Address Register
DMCTR1 .set 07h ;Channel 1 Element Count Register
DMSFC1 .set 08h ;Channel 1 Sync Select and Frame Count Register
DMMCR1 .set 09h ;Channel 1 Transfer Mode Control Register
DMSRC2 .set 0Ah ;Channel 2 Source Address Register
DMDST2 .set 0Bh ;Channel 2 Destination Address Register
DMCTR2 .set 0Ch ;Channel 2 Element Count Register
DMSFC2 .set 0Dh ;Channel 2 Sync Select and Frame Count Register
DMMCR2 .set 0Eh ;Channel 2 Transfer Mode Control Register
DMSRC3 .set 0Fh ;Channel 3 Source Address Register
DMDST3 .set 10h ;Channel 3 Destination Address Register
DMCTR3 .set 11h ;Channel 3 Element Count Register
DMSFC3 .set 12h ;Channel 3 Sync Select and Frame Count Register
DMMCR3 .set 13h ;Channel 3 Transfer Mode Control Register
DMSRC4 .set 14h ;Channel 4 Source Address Register
DMDST4 .set 15h ;Channel 4 Destination Address Register
DMCTR4 .set 16h ;Channel 4 Element Count Register
DMSFC4 .set 17h ;Channel 4 Sync Select and Frame Count Register
DMMCR4 .set 18h ;Channel 4 Transfer Mode Control Register
DMSRC5 .set 19h ;Channel 5 Source Address Register
DMDST5 .set 1Ah ;Channel 5 Destination Address Register
DMCTR5 .set 1Bh ;Channel 5 Element Count Register
DMSFC5 .set 1Ch ;Channel 5 Sync Select and Frame Count Register
DMMCR5 .set 1Dh ;Channel 5 Transfer Mode Control Register
DMSRCP .set 1Eh ;Source Program Page Address
DMDSTP .set 1Fh ;Destination Program Page Address
DMIDX0 .set 20h ;Element Address Index Register 0
DMIDX1 .set 21h ;Element Address Index Register 1
DMFRI0 .set 22h ;Frame Address Index Register 0
DMFRI1 .set 23h ;Frame Address Index Register 1
DMGSA .set 24h ;Global Source Address Reload Register
DMGDA .set 25h ;Global Destination Address Reload Register
DMGCR .set 26h ;Global Element Count Reload Register
DMGFR .set 27h ;Global Frame Count Reload Register
```

### 例3-16. プログラム・メモリーからデータ・メモリーへの転送(自動インクリメントなし)

この例では、DMAを通じてプログラム空間からデータ空間へデータ・ブロックを転送します。DMAコントローラは1000hのシングル・ワード転送を行い、各エレメントの転送ごとに転送元/転送先の両方のアドレスを1ずつインクリメントするように設定されています。この転送は、同期イベントに関連付けていません(フリー・ラン)。ブロック転送が完了すると(自動初期化はオフ)、DMAチャンネルはディセーブルになります。

|         |                     |
|---------|---------------------|
| 転送モード   | マルチフレーム・モード         |
| 転送元アドレス | 18000h(プログラム空間)     |
| 転送先アドレス | 03000h(データ空間)       |
| 転送サイズ   | 1000hシングル(16ビット)ワード |
| 同期イベント  | なし(フリー・ラン)          |
| 使用チャンネル | DMAチャンネル0           |

```
*****
stm    DMSRCP,DMSA          ;set source program page to 1
stm    #1h,DMSDN
stm    DMSRC0,DMSA          ;set source program address to 8000
stm    #8000h,DMSDN         ; (lower 16-bit of 18000h)
stm    DMDST0,DMSA          ;set destination address to 3000
stm    #3000h,DMSDN
stm    DMCTR0 ,DMSA          ;set for 1000h transfers
stm    #(1000h-1) ,DMSDN
stm    DMSFC0 ,DMSA
stm    #0000000000000000b ,DMSDN
        ;0000~ (DSYN)          No sync event
        ;~~~0~ (DBLW)          Single-word mode
        ;~~~~000~ Reserved
        ;~~~~~00000000 (Frame Count)  Frame Count=0h (one frame)
stm    DMMCR0 ,DMSA
stm    #0000000100000101b ,DMSDN
        ;0~ (AUTOINIT)  Autoinitialization disabled
        ;~0~ (DINM)      Interrupts masked
        ;~~0~ (IMOD)      N/A
        ;~~~0~ (CTMOD)    Multi-frame mode
        ;~~~~0~ Reserved
        ;~~~~~001~ (SIND)  Post increment source address
        ;~~~~~00~ (DMS)    Source in program space
        ;~~~~~0~ Reserved
        ;~~~~~001~ (DIND)  Post increment destination address
        ;~~~~~01 (DMD)     Destination in data space

stm    #0000000100000001b ,DMPREC
        ;0~ (FREE)        DMA stops on emulation stop
        ;~0~ Reserved
        ;~~0~ (DPRC[5])    Channel 5 low priority
        ;~~~0~ (DPRC[4])    Channel 4 low priority
        ;~~~~0~ (DPRC[3])    Channel 3 low priority
        ;~~~~~0~ (DPRC[2])    Channel 2 low priority
        ;~~~~~0~ (DPRC[1])    Channel 1 low priority
        ;~~~~~1~ (DPRC[0])    Channel 0 high priority
        ;~~~~~00~ (INTOSEL)  N/A
        ;~~~~~0~ (DE[5])     Channel 5 disabled
        ;~~~~~0~ (DE[4])     Channel 4 disabled
        ;~~~~~0~ (DE[3])     Channel 3 disabled
        ;~~~~~0~ (DE[2])     Channel 2 disabled
        ;~~~~~0~ (DE[1])     Channel 1 disabled
        ;~~~~~1 (DE[0])      Channel 0 enabled
*****
```

### 例3-17. プログラム・メモリーからデータ・メモリーへの転送(自動インクリメントあり)

この例では、前の例と同じ転送を行いますが、自動インクリメント機能を持つサブアドレッシング(DSMBAIレジスタ)を用いて設定しています。このサブアドレッシングを使用すると、1命令でチャンネル・コンテキスト・レジスタを設定でき、時間とメモリーの節約になります。

|         |                     |
|---------|---------------------|
| 転送モード   | マルチフレーム・モード         |
| 転送元アドレス | 18000h(プログラム空間)     |
| 転送先アドレス | 03000h(データ空間)       |
| 転送サイズ   | 1000hシングル(16ビット)ワード |
| 同期イベント  | なし(フリー・ラン)          |
| 使用チャンネル | DMAチャンネル番号0         |

\*\*\*\*\*

```

stm    DMSRCP,DMSA      ;set source program page to 1
stm    #1h,DMSDN

stm    DMSRC0,DMSA      ;set source program address to 8000 and
stm    #8000h,DMSDI     ; point DMSA to next sub-address (DMDST0)

stm    #3000h,DMSDI     ;set destination address to 3000 and
                        ; point DMSA to next sub-address (DMCTR0)

stm    #(1000h-1),DMSDI ;set for 1000h transfers and point
                        ; DMSA to next sub-address (DMSFC0)

stm    #00000h,DMSDI    ;configure DMSFC0 and point DMSA
                        ; to next sub-address (DMMCR0)

stm    #00105h,DMSDI    ;configure DMMCR0 and point DMSA
                        ; to next sub-address (DMSRC0)

stm    #00101h,DMPREC   ;configure DMPREC

```

\*\*\*\*\*

例3-18. プログラム・メモリーからデータ・メモリーへの転送(自動初期化あり)

この例では、DMAを通じてプログラム空間からデータ空間へデータ・ブロックを転送します。DMAコントローラは1ブロックあたり1000hのシングル・ワード転送を行い、各エレメント転送ごとに、転送元/転送先の両方のアドレスを1ずつインクリメントするように設定されています。この転送は、同期イベントに関連付けていません(フリー・ラン)。最初のブロック転送が完了すると、グローバル・リロード・レジスタ(DMGSA、DMGDA、DMGCR、DMGFR)のデータを使って、DMAチャンネルが自動的に初期化され、再度転送が開始されます。グローバル・リロード・レジスタは転送先を最初のブロック転送時の03000hではなく、データ空間の02000hに指定します。また、グローバル・リロード・レジスタを使って最初のブロック転送と同じ条件で自動初期化することも可能です。

|                    |                                           |
|--------------------|-------------------------------------------|
| 転送モード              | マルチフレーム・モード                               |
| 最初の転送元アドレス         | 18000h(プログラム空間)                           |
| 最初の転送先アドレス         | 03000h(データ空間)                             |
| 最初の転送サイズ           | 1000hシングル(16ビット)ワード<br>(1フレーム、1000hエレメント) |
| 自動初期化による転送元アドレス    | 18000h(プログラム空間)                           |
| 自動初期化による転送先アドレス    | 02000h(データ空間)                             |
| 自動初期化によるエレメント・カウント | 1000hシングル(16ビット)ワード                       |
| 自動初期化によるフレーム・カウント  | 000h (1フレーム)                              |
| 同期イベント             | なし(フリー・ラン)                                |
| 使用チャンネル            | DMAチャンネル4                                 |

\*\*\*\*\*

```

stm    DMSRCP,DMSA          ;set source program page to 1
stm    #1h,DMSDN
stm    DMSRC4,DMSA          ;set source program address to 18000h
stm    #8000h,DMSDN
stm    DMDST4,DMSA          ;set destination address to 3000h
stm    #3000h,DMSDN
stm    DMCTR4 ,DMSA          ;set for 1000h transfers
stm    #(1000h-1) ,DMSDN
stm    DMSFC4 ,DMSA
stm    #0000000000000000b ,DMSDN
;0000~ (DSYN)                No sync event
;~~~~0~ (DBLW)                Single-word mode
;~~~~000~ Reserved
;~~~~~00000000 (Frame Count) Frame Count = 0h (one frame)

```



```

stm    DMMCR4 ,DMSA
stm    #1000000100000101b ,DMSDN
;1~~~~~ (AUTOINIT) Autoinitialization enabled
;~0~~~~~ (DINM) Interrupts masked
;~~0~~~~~ (IMOD) N/A
;~~~0~~~~~ (CTMOD) Multi-frame mode
;~~~~0~~~~~ Reserved
;~~~~001~~~~~ (SIND) Post increment source address
;~~~~~00~~~~~ (DMS) Source in program space
;~~~~~0~~~~~ Reserved
;~~~~~001~~ (DIND) Post increment destination address
;~~~~~01~~ (DMD) Destination in data space

stm    DMGSA,DMSA          ;set global source address to 8000h
stm    #8000h,DMSDN        ; (lower 16-bits of 18000h in program space)
stm    DMGDA,DMSA          ;set global destination address to 2000h
stm    #2000h,DMSDN
stm    DMGCR,DMSA          ;set global element count to move 1000h words
stm    #(1000h-1),DMSDN
stm    DMGFR,DMSA          ;set global frame count to 0h

stm    #0000000000000000b,DMSDN
;00000000~~~~~ Reserved
;~~~~~00000000 Global Frame count

stm    #00010000000010000b ,DMPREC
;0~~~~~ (FREE) DMA stops on emulation stop
;~0~~~~~ Reserved
;~~0~~~~~ (DPRC[5]) Channel 5 low priority
;~~~1~~~~~ (DPRC[4]) Channel 4 high priority
;~~~~0~~~~~ (DPRC[3]) Channel 3 low priority
;~~~~~0~~~~~ (DPRC[2]) Channel 2 low priority
;~~~~~0~~~~~ (DPRC[1]) Channel 1 low priority
;~~~~~0~~~~~ (DPRC[0]) Channel 0 low priority
;~~~~~00~~~~~ (INTOSEL) N/A
;~~~~~0~~~~~ (DE[5]) Channel 5 disabled
;~~~~~1~~~~~ (DE[4]) Channel 4 enabled
;~~~~~0~~~~~ (DE[3]) Channel 3 disabled
;~~~~~0~~~~~ (DE[2]) Channel 2 disabled
;~~~~~0~~~~~ (DE[1]) Channel 1 disabled
;~~~~~0~~~~~ (DE[0]) Channel 0 disabled
*****

```

### 例3-19. ABUモードでのMcBSPデータ転送

この例では、McBSPで受信された16ビット・ワードを、DMAコントローラを使ってデータ空間に転送します。転送先ではサーキュラ・バッファを設定し、各転送ごとに転送先のバッファ・アドレスが1インクリメントされます。ABUモードは、サーキュラ・バッファを実行するのに用いられるのです。ABUモードでは、エレメント・カウント・レジスタがバッファ・サイズを表し、フレーム・カウントは使われません。この例では、100hのバッファが一杯になったときにCPUに割り込みを発生させる設定のみが示されています。この割り込みに対する割り込みサービス・ルーチンは、この例には記載されていません。

|          |                        |
|----------|------------------------|
| 転送モード    | ABU(ノン・デクリメント)モード      |
| 転送元アドレス  | McBSP0データ受信レジスタ(DRR10) |
| 転送先バッファ  | 03000h~030FFh(データ空間)   |
| バッファ・サイズ | 100hシングル(16ビット)ワード     |
| 同期イベント   | McBSP0受信イベント           |
| 使用チャンネル  | DMAチャンネル1              |

```

*****
stm    DMSRC1,DMSA          ;set source address to DRR10
stm    DRR1_0,DMSDN
stm    DMDST1,DMSA          ;set destination address to 3000
stm    #3000h,DMSDN
stm    DMCTR1 ,DMSA          ;set buffer size to 100h words
stm    #100h ,DMSDN
stm    DMSFC1 ,DMSA
stm    #0001000000000000b ,DMSDN
        ;0001~ (DSYN)          McBSP0 receive sync event
        ;~~~0~ (DBLW)          Single-word mode
        ;~~~~000~ Reserved
        ;~~~~~00000000 (Frame Count) Frame count is not
        ;                                     relevant in ABU mode

stm    DMMCR1 ,DMSA
stm    #0101000001001101b ,DMSDN
        ;0~ (AUTOINIT) Autoinitialization disabled
        ;~1~ (DINM) DMA Interrupts enabled
        ;~~0~ (IMOD) Interrupt at full buffer
        ;~~~1~ (CTMOD) ABU (non-decrement) mode
        ;~~~~0~ Reserved
        ;~~~~~000~ (SIND) No modify on source address (DRR10)
        ;~~~~~01~ (DMS) Source in data space
        ;~~~~~0~ Reserved
        ;~~~~~011~ (DIND) Post increment destination address
        ;                                     with DMIDX0
        ;~~~~~01 (DMD) Destination in data space

stm    DMIDX0,DMSA          ;set element address index to +1
stm    #0001h,DMSDN

stm    #0000001000000010b ,DMPREC
        ;0~ (FREE) DMA stops on emulation stop
        ;~0~ Reserved
        ;~~0~ (DPRC[5]) Channel 5 low priority
        ;~~~0~ (DPRC[4]) Channel 4 low priority
        ;~~~~0~ (DPRC[3]) Channel 3 low priority
        ;~~~~~0~ (DPRC[2]) Channel 2 low priority
        ;~~~~~1~ (DPRC[1]) Channel 1 high priority
        ;~~~~~0~ (DPRC[0]) Channel 0 low priority
        ;~~~~~00~ (INTOSEL) N/A
        ;~~~~~0~ (DE[5]) Channel 5 disabled
        ;~~~~~0~ (DE[4]) Channel 4 disabled
        ;~~~~~0~ (DE[3]) Channel 3 disabled
        ;~~~~~0~ (DE[2]) Channel 2 disabled
        ;~~~~~1~ (DE[1]) Channel 1 enabled
        ;~~~~~0~ (DE[0]) Channel 0 disabled
*****

```

例3-20. ダブルワード・モードでのMcBSPデータ転送

この例では、McBSPで受信された32ビット・ワードをDMAのマルチフレーム・モードでデータ空間に転送します。DMAコントローラは、DRR20とDRR10を自動的にリードし、データ空間03000hからの連続領域にデータを転送するよう設定されています。

3

|         |                              |
|---------|------------------------------|
| 転送モード   | マルチフレーム・モード                  |
| 転送元アドレス | McBSP0データ受信レジスタ(DRR20/DRR10) |
| 転送先アドレス | 03000h(データ空間)                |
| 転送サイズ   | 50hダブル(32ビット)ワード             |
| 同期イベント  | McBSP0受信イベント                 |
| 使用チャンネル | DMAチャンネル1                    |

```
*****
stm    DMSRC1,DMSA          ;set source address to DRR20
stm    DRR2_0,DMSDN
stm    DMDST1,DMSA          ;set destination address to 3000
stm    #3000h,DMSDN
stm    DMCTR1 ,DMSA          ;set buffer size to 50h words
stm    #050h-1,DMSDN

stm    DMSFC1 ,DMSA
stm    #0001100000000000b ,DMSDN
        ;0001~ (DSYN)          McBSP0 receive sync event
        ;~~~1~ (DBLW)          Double-word mode
        ;~~~~000~ Reserved
        ;~~~~~00000000 (Frame Count) Single Frame

stm    DMMCR1 ,DMSA
stm    #0100000001001101b ,DMSDN
        ;0~ (AUTOINIT) Autoinitialization disabled
        ;~1~ (DINM) DMA Interrupts enabled
        ;~~0~ (IMOD) Interrupt at complete block transfer
        ;~~~0~ (CTMOD) Multi-frame mode
        ;~~~~0~ Reserved
        ;~~~~~000~ (SIND) No modify on source address (DRR20)
        ;~~~~~01~ (DMS) Source in data space
        ;~~~~~0~ Reserved
        ;~~~~~011~ (DIND) Post increment destination address
        ; with DMIDX0
        ;~~~~~01 (DMD) Destination in data space
stm    DMIDX0,DMSA          ;set element address index to +1
stm    #0001h,DMSDN

stm    #00000010000000010b ,DMPREC
        ;0~ (FREE) DMA stops on emulation stop
        ;~0~ Reserved
        ;~~0~ (DPRC[5]) Channel 5 low priority
        ;~~~0~ (DPRC[4]) Channel 4 low priority
        ;~~~~0~ (DPRC[3]) Channel 3 low priority
        ;~~~~~0~ (DPRC[2]) Channel 2 low priority
        ;~~~~~1~ (DPRC[1]) Channel 1 high priority
        ;~~~~~0~ (DPRC[0]) Channel 0 low priority
        ;~~~~~00~ (INTOSEL) N/A
        ;~~~~~0~ (DE[5]) Channel 5 disabled
        ;~~~~~0~ (DE[4]) Channel 4 disabled
        ;~~~~~0~ (DE[3]) Channel 3 disabled
        ;~~~~~0~ (DE[2]) Channel 2 disabled
        ;~~~~~1~ (DE[1]) Channel 1 enabled
        ;~~~~~0~ (DE[0]) Channel 0 disabled
*****
```

### 例3-21. データのソートを伴うMcBSPからデータ・メモリーへの転送

この例は、3-23ページの図3-5に示したデータ・ソートの例です。ここでは、McBSPで受信されたシングル・ワードを、03000h番地から始まるデータ・メモリーに転送します。転送されたデータは、4個の要素を含む4つのフレームで構成されています。DMAでは、要素・インデックス・レジスタ(DMIDX0)とフレーム・インデックス・レジスタ(DMFRI0)で転送先アドレスを変更することによってデータのソートを行います。この結果、ストアされたデータはフレーム番号ではなく要素番号順にソートされます。

|         |                        |
|---------|------------------------|
| 転送モード   | マルチフレーム・モード            |
| 転送元アドレス | McBSP0データ受信レジスタ(DRR10) |
| 転送先アドレス | 03000h(データ空間)          |
| 転送サイズ   | 10hシングル(16ビット)ワード      |
| 同期イベント  | McBSP0受信イベント           |
| 使用チャネル  | DMAチャネル1               |

```
*****
stm    DMSRC1,DMSA          ;set source address to DRR10
stm    DRR1_0,DMSDN
stm    DMDST1,DMSA          ;set destination address to 3000
stm    #3000h,DMSDN
stm    DMCTR1 ,DMSA          ;set elements per frame to 4h
stm    #0004h-1 ,DMSDN
stm    DMSFC1 ,DMSA
stm    #0001000000000011b ,DMSDN
;0001~ (DSYN)          McBSP0 receive sync event
;~~~0~ (DBLW)          Single-word mode
;~~~~000~ Reserved
;~~~~~00000011 (Frame Count)  Frames per block = 4h-1
stm    DMMCR1 ,DMSA
stm    #0100000001010101b ,DMSDN
;0~ (AUTOINIT) Autoinitialization disabled
;~1~ (DINM) DMA Interrupts enabled
;~~0~ (IMOD) Interrupt at complete block transfer
;~~~0~ (CTMOD) Multi-frame mode
;~~~~0~ Reserved
;~~~~~000~ (SIND) No modify on source address (DRR10)
;~~~~~01~ (DMS) Source in data space
;~~~~~0~ Reserved
;~~~~~101~ (DIND) Post increment destination address
; with DMIDX0 and DMFRI0
;~~~~~01 (DMD) Destination in data space
stm    DMIDX0,DMSA          ;set element address index to +4
stm    #0004h,DMSDN

stm    DMFRI0,DMSA          ;set frame address index to -11
stm    #0FFF5h,DMSDN

stm    #0000001000000010b ,DMPREC
;0~ (FREE) DMA stops on emulation stop
;~0~ Reserved
;~~0~ (DPRC[5]) Channel 5 low priority
;~~~0~ (DPRC[4]) Channel 4 low priority
;~~~~0~ (DPRC[3]) Channel 3 low priority
;~~~~~0~ (DPRC[2]) Channel 2 low priority
;~~~~~1~ (DPRC[1]) Channel 1 high priority
;~~~~~0~ (DPRC[0]) Channel 0 low priority
;~~~~~00~ (INTOSEL) N/A
;~~~~~0~ (DE[5]) Channel 5 disabled
;~~~~~0~ (DE[4]) Channel 4 disabled
;~~~~~0~ (DE[3]) Channel 3 disabled
;~~~~~0~ (DE[2]) Channel 2 disabled
;~~~~~1~ (DE[1]) Channel 1 enabled
;~~~~~0~ (DE[0]) Channel 0 disabled
*****
```





# 拡張8ビット・ホスト・ポート・インターフェイス (HPI-8)

拡張8ビット・ホスト・ポート・インターフェイス(HPI-8)は、ホスト・デバイスやホスト・プロセッサとC54xとのインターフェイスをとるために設計された、スタンダード8ビットHPIを改良したものです。

HPI-8は、TI DSPのC54xファミリーの一部のデバイスで使用できます。

| Topic                                            | Page |
|--------------------------------------------------|------|
| 4.1 拡張8ビット・ホスト・ポート・インターフェイス(HPI-8)の概要 .....      | 4-2  |
| 4.2 HPI-8の基本機能について .....                         | 4-4  |
| 4.3 HPI-8の動作について .....                           | 4-6  |
| 4.4 ホストからHPI-8へのリード/ライト・アクセス .....               | 4-14 |
| 4.5 DSPINTとHINTの動作 .....                         | 4-23 |
| 4.6 クロック・モード変更時のHPI-8転送に関する考慮事項 .....            | 4-24 |
| 4.7 IDLE使用時の考慮事項 .....                           | 4-26 |
| 4.8 リセット時におけるHPI-8の動作 .....                      | 4-28 |
| 4.9 HPI-8データ・ピンの汎用I/Oピンとしての使用(C5410では使用不可) ..... | 4-30 |

## 4.1 拡張8ビット・ホスト・ポート・インターフェイス(HPI-8)の概要

HPI-8は、ホスト・デバイスやホスト・プロセッサと'C54xを接続するための8ビットの平行・ポートです。'C54xとホスト・デバイス間では、'C54xのオンチップRAMを介してデータがやり取りされます。

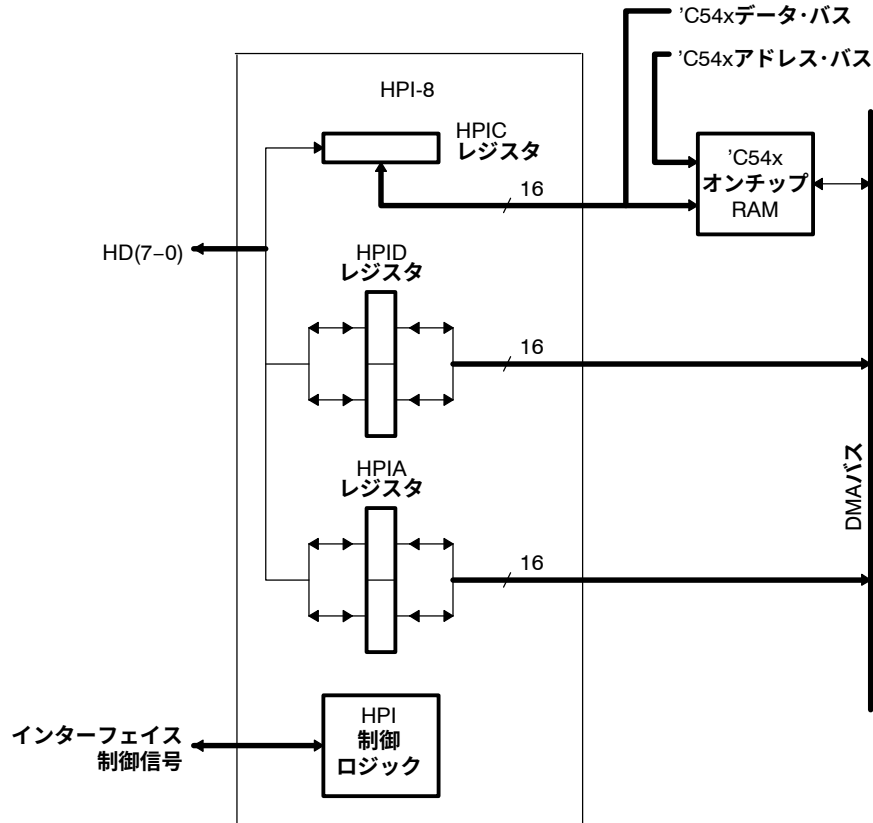
表4-1に、拡張8ビットHPIとスタンダード8ビットHPIのおもな違いを示します。

表4-1. 拡張8ビットHPIとスタンダード8ビットHPIのおもな違い

| 拡張8ビットHPI(HPI-8)                                                 | スタンダード8ビットHPI                   |
|------------------------------------------------------------------|---------------------------------|
| <input type="checkbox"/> オンチップRAMの全域にアクセス可能                      | オンチップRAMのうち特定の2Kワードの領域にのみアクセス可能 |
| <input type="checkbox"/> ホスト・アクセスは常に'C54xのクロックと同期される(ホスト専用モードなし) | ホスト専用モードにより非同期のホスト・アクセスが可能      |
| <input type="checkbox"/> ホストと'C54xの両方がオンチップRAMに常時アクセス可能          | ホスト専用モードによりホストがRAMへのアクセス権を占有可能  |

HPI-8はスレーブとして機能し、ホスト・プロセッサが'C54xのオンチップ・メモリーにアクセスできるようにします。このインターフェイスは8ビットの双方向データ・バスと各種制御信号で構成されています。16ビット転送は、ハイ・バイトかロー・バイトかを指定するHBIL入力を使った2回のアクセスで行われます。ホストは、'C54xからは直接アクセスできない専用のアドレス・レジスタとデータ・レジスタを使ってHPI-8と通信します。ホストと'C54xの両方がアクセス可能なHPI制御レジスタには、プロトコルの設定と通信制御(ハンドシェーク)を行うためのビットが含まれています。図4-1に、HPI-8の簡単なブロック図を示します。

図4-1. ホスト・ポート・インターフェイスのブロック図



4

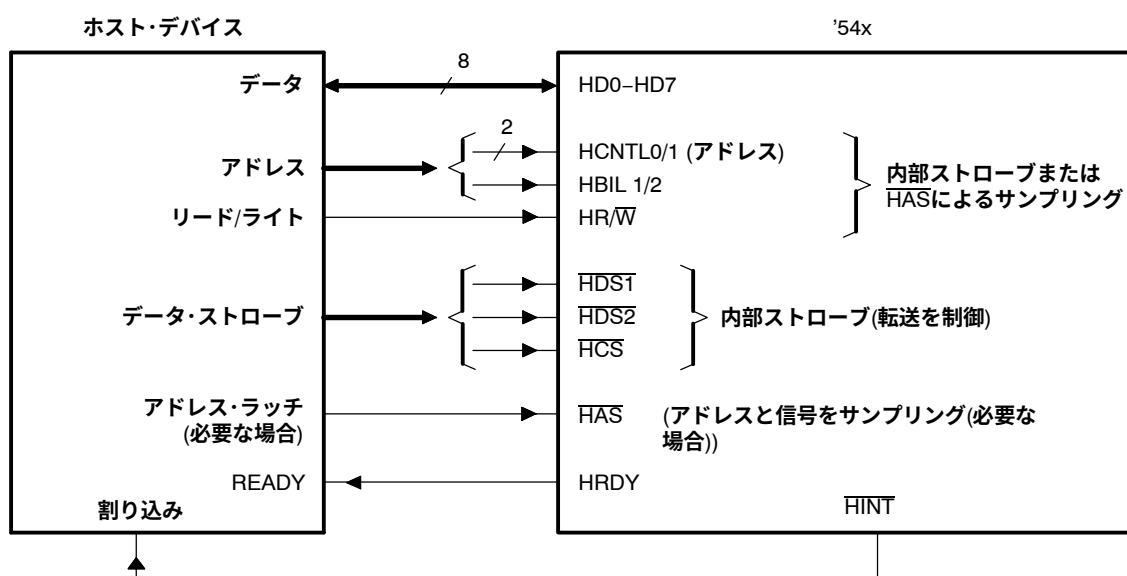
HPI-8は8ビットの外部インターフェイスでありながら、'C54xに16ビットのデータを提供します。連続的に転送される2つのバイトは自動的に結合されて16ビット・ワードになります。ホスト・デバイスがHPI-8のレジスタに対してデータ転送を行うと、HPI-8の制御ロジックが自動的に'C54xの内部RAMへアクセスします。この後、'C54xがメモリー空間内に書き込まれたデータにアクセスできるようになります。

'C54xとホストはどちらも、'C54xのオンチップRAM全域にアクセスすることができます。ホストからアクセスを行うには'C54xのクロックがアクティブ状態でなければならず、また、'C54xデバイスがリセット中の場合にはHPI-8は機能しません(ただし、'C5410の場合は例外です。詳細については、4-28ページの第4.8.2節「リセット時のHPI-8へのアクセス('C5410のみ)」を参照してください)。オンチップRAMへのアクセスを正しく処理するため、ホスト・アクセスは'C54xのクロックと内部で同期が取られます。'C54xとホストの両方が同じメモリー位置にアクセスして、'C54xとホスト・サイクル間で競合が発生した場合は、ホストからのアクセスが優先され、'C54xのCPUは1クロック・サイクル待たされます。

## 4.2 HPI-8の基本機能について

HPI-8の外部インターフェイスは各種のホスト・デバイスに接続でき、その際に追加ロジックはほとんど、あるいは全く必要としません。ホストとのデータ交換には、8ビット・データ・バス(HD0～HD7)が使われます。2つの制御入力(HCNTL0とHCNTL1)は、HPI-8のどの内部レジスタにアクセスするかを指定します。これらの入力信号とHBILは通常、ホストのアドレス・バス・ビットまたは同等の機能を持つビットによって制御されます。図4-2に、HPI-8とホスト・デバイス間の簡単な接続図を示します。

図4-2. 一般的なシステム・ブロック図



ホストはHCNTL0/1入力を使って、HPI制御レジスタ(HPIC)、HPIアドレス・レジスタ(HPIA)('C54xのRAMを指すポインタとして機能)、HPIデータ・レジスタ(HPID)のどれにアクセスするかを指定します。'C54xは16ビット・ワード構造のため、HPI-8での転送はすべて連続2バイトで構成する必要があります。第1バイトと第2バイトのどちらを転送するかは、専用のHBILピンによって指定されます。なお、どちらのバイトが16ビット・ワードの上位バイトに置かれるかは、内部制御レジスタのビットによって決まります。

また、HPIDレジスタは自動アドレス・インクリメント機能を使ってアクセスすることができます。自動インクリメント機能を使用すれば、連続して配置されているワードのリード/ライトが容易になります。自動インクリメント・モードにすると、連続転送中にHPIAレジスタが自動的にインクリメントされます。この機能の詳細については、4-11ページの第4.3.3節「アドレスの自動インクリメント」を参照してください。

HPI-8には、ソフトウェア上のハンドシェークを容易にするための割り込みロジックが組み込まれています。ホストはHPIC内の専用ビットにライトすることで、'C54xのCPUに割り込みを発生させることができます。同様に、'C54xもHINT出力ピンを使ってホストに割り込みを発生させることができます。'C54xがHPIC内の専用ビットにライトすると、HINT出力がドライブされます。ホストはまた、HPICレジスタのHINTビットにライトすることで、HINTピンの認識とクリアを行うこともできます。

表4-2では、HPI-8がホスト・デバイスと'C54xの通信に利用する3つのレジスタについて説明しています。

表4-2. HPI-8レジスタの説明

| 名前   | 'C54xアドレス | 説明                                                                                                                           |
|------|-----------|------------------------------------------------------------------------------------------------------------------------------|
| HPIA | —         | HPIアドレス・レジスタ。直接アクセスできるのはホストだけです。'C54xオンチップRAM内のアドレスを保持し、オンチップRAM内のそのアドレスが示す部分へのアクセスが発生します。                                   |
| HPIC | 002Ch     | HPI制御レジスタ。ホストと'C54xのいずれかが直接アクセスできます。ここには、HPI-8の動作に関する制御ビットとステータス・ビットが含まれています。                                                |
| HPID | —         | HPIデータ・レジスタ。直接アクセスできるのはホストだけです。リード・アクセスが行われる場合は、'C54xのオンチップ・メモリーからリードされたデータを保持し、ライト・アクセスが行われる場合は、オンチップ・メモリーにライトされるデータを保持します。 |

各種のストローブ信号(データ・ストローブ( $\overline{\text{HDS1}}$ と $\overline{\text{HDS2}}$ )、リード/ライト・ストローブ( $\text{HR}/\overline{\text{W}}$ )、アドレス・ストローブ( $\overline{\text{HAS}}$ )など)を使用すれば、HPI-8をさまざまなホスト・デバイスに接続できます。

HPIのレディー・ピン( $\text{HRDY}$ )により、HPI-8のアクセス速度に合わせてホストのアクセス速度を自動調整できます。 $\text{HRDY}$ ピンを使用すれば、レディー入力を持つホストにウエートステートを挿入できます。このような制御により、ホストのバス・サイクル・タイムがHPI-8のアクセス速度よりも高速な場合に、ホスト・アクセスを遅延処理することができます。

これらの機能の組み合わせにより、業界標準の幅広いホスト・デバイスに対して、柔軟かつ効率的に接続することができます。

### 4.3 HPI-8の動作について

この節では、HPI-8の外部インターフェイス・ピンの各機能について詳しく説明します。また、制御レジスタのビット機能とHPI-8メモリー・マップについても説明します。インターフェイスのタイミング、HPI-8の初期設定、リード/ライト・シーケンスについては、4-14ページの第4.4節「ホストからHPI-8へのリード/ライト・アクセス」を参照してください。

HPI-8の外部インターフェイス信号は、各種ホスト・デバイスとのインターフェイスに柔軟に対応します。デバイスのデータ・ストロブが単一か複数か、あるいはアドレス・ラッチ・イネーブル(ALE)信号があるか否かに関わらず、さまざまなデバイスをHPI-8に容易に接続することができます。

表4-3では、HPI-8の外部インターフェイス・ピンそれぞれについて詳しく説明しています。

表4-3. HPI-8の各信号の名称とその機能

| HPIピン                   | ホスト・ピン                                      | 状態 <sup>†</sup> | 信号の機能                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------------|---------------------------------------------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $\overline{\text{HAS}}$ | アドレス・ラッチ・イネーブル(ALE)、アドレス・ストロブ、または未使用(ハイに接続) | I               | アドレス・ストロブ入力。アドレスおよびデータ・バスを共有しているホストは、 $\overline{\text{HAS}}$ をALEピン(またはそれと同等のピン)に接続します。HBIL、HCNTL0/1、HR/ $\overline{\text{W}}$ は $\overline{\text{HAS}}$ の立ち下がりエッジでラッチされます。 $\overline{\text{HAS}}$ を使用する場合は、HCS、HDS1、HDS2の最も遅い信号よりも必ず先行してアドレスおよび条件制御信号をラッチします(HPI-8のタイミングの仕様については、'C54xのデータ・シートを参照してください)。アドレスおよびデータ・バスが独立したホストは、 $\overline{\text{HAS}}$ をハイに接続します。この場合は、 $\overline{\text{HAS}}$ がインアクティブ(ハイ)状態で、HDS1、HDS2、HCSの中で最も遅い信号の立ち下がりエッジによってHBIL、HCNTL0/1、HR/ $\overline{\text{W}}$ がラッチされます。 |
| HBIL                    | アドレスまたは制御ライン                                | I               | バイト識別入力。第1バイトと第2バイトのどちらが転送されるかを識別します(上位か下位かを識別するわけではありません。この節で後述しますが、上位か下位かはHPICレジスタのBOBビットによって指定されます)。第1バイトの場合はHBILがローになり、第2バイトの場合はハイになります。                                                                                                                                                                                                                                                                                                                                                              |
| HCNTL0、HCNTL1           | アドレスまたは制御ライン                                | I               | ホスト制御入力。HPIAレジスタ、HPIデータ・ラッチ(オプションでアドレス・インクリメントあり)、HPICレジスタのいずれにホストがアクセスするかを選択します。                                                                                                                                                                                                                                                                                                                                                                                                                         |
| $\overline{\text{HCS}}$ | アドレスまたは制御ライン                                | I               | チップ・セレクト。HPI-8のイネーブル入力として機能します。アクセス中は必ずローにしますが、それ以外の時でもローのままでかまいません。通常、 $\overline{\text{HCS}}$ はHDS1とHDS2に先行しますが、 $\overline{\text{HAS}}$ が使われず、HDS1とHDS2のどちらかがすでにローの場合は、この信号によってHCNTL0/1、HR/ $\overline{\text{W}}$ 、HBILをサンプルします(詳細については、この節で後述します)。4-8ページの図4-3「HPIストロブと選択ロジック」は、HCS、HDS1、HDS2の各入力の等価回路を示しています。                                                                                                                                                                                         |

<sup>†</sup> I=入力、O=出力、Z=ハイ・インピーダンス

表4-3. HPI-8の各信号の名称とその機能 (続き)

| HPIピン                                               | ホスト・ピン                                 | 状態 <sup>†</sup> | 信号の機能                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------------------------------------|----------------------------------------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| HD0~HD7                                             | データ・バス                                 | I/O/Z           | パラレル双方向3ステート・データ・バス。HPIがインアクティブ( $\overline{\text{HDSx}}$ OR $\overline{\text{HCS}}=1$ )の場合か、EMU1/OFFがアクティブ(ロー)の場合は、HD7(MSB)~HD0(LSB)出力はハイ・インピーダンス状態になります。HPI-8がディセーブルの場合は、これらのピンを汎用入力/出力にも使用できます。この機能の詳細については、4-30ページの第4.9節「HPI-8データ・ピンの汎用I/Oピンとしての使用」を参照してください。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| $\overline{\text{HDS1}}$ 、 $\overline{\text{HDS2}}$ | リード・ストローブおよびライト・ストローブまたはデータ・ストローブ      | I               | データ・ストローブ入力。ホストのアクセス・サイクル中のデータ転送を制御します。また、 $\overline{\text{HAS}}$ が使われていない場合で、 $\overline{\text{HCS}}$ がすでにローになっている時(通常動作の場合)は、これらの入力によってHBIL、HCNTL0/1、 $\text{HR}/\overline{\text{W}}$ がサンプルされます。リード・ストローブとライト・ストローブが独立のホストは、これらのストローブを $\overline{\text{HDS1}}$ と $\overline{\text{HDS2}}$ のどちらかに接続します。一方、単一のデータ・ストローブしか持たないホストは、ストローブを $\overline{\text{HDS1}}$ と $\overline{\text{HDS2}}$ のいずれかに接続し、未使用のピンをハイに接続します。 $\overline{\text{HDS}}$ の接続状態に関わらず、転送方向を決定するために $\text{HR}/\overline{\text{W}}$ が必要となります。 $\overline{\text{HDS1}}$ と $\overline{\text{HDS2}}$ は内部でXNOR処理されるため、ハイ・アクティブのデータ・ストローブを持つホストは、そのストローブを $\overline{\text{HDS}}$ 入力のどちらか一方に接続し、他方の $\overline{\text{HDS}}$ 入力をローに接続します。4-8ページの図4-3は、 $\overline{\text{HDS1}}$ 、 $\overline{\text{HDS2}}$ 、 $\overline{\text{HCS}}$ の各入力の等価回路を示しています。 |
| HINT                                                | ホスト割り込み出力                              | O/Z             | ホスト割り込み出力。HPIC内のHINTビットによって制御されます。 $^{\circ}\text{C54x}$ のリセット中はハイにドライブされます。EMU1/OFFがアクティブ(ロー)の場合はハイ・インピーダンスになります。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| HRDY                                                | 非同期レディー                                | O/Z             | HPIレディー出力。ハイの場合には、HPI-8が転送を実行できる状態であることを示します。ローの場合は、HPI-8が前の転送の内部処理を実行中のためビジー状態であることを示します。EMU1/OFFがアクティブ(ロー)の場合はハイ・インピーダンスになります。なお、HRDYは $\overline{\text{HCS}}$ によってイネーブルになります(つまり、 $\overline{\text{HCS}}$ がハイのときにはHRDYも常にハイになります)。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| $\text{HR}/\overline{\text{W}}$                     | リード/ライト・ストローブ、アドレス・ライン、またはアドレス/データ共用バス | I               | リード/ライト入力。ホストはHPI-8からデータをリードする場合は $\text{HR}/\overline{\text{W}}$ をハイにドライブし、HPI-8にデータをライトする場合はローにドライブする必要があります。リード/ライト・ストローブを持たないホストは、この信号をアドレス・ラインより制御できます。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

<sup>†</sup> I=入力、O=出力、Z=ハイ・インピーダンス

$\overline{\text{HCS}}$ 入力は主にHPI-8のイネーブル入力として機能し、 $\overline{\text{HDS1}}$ 信号と $\overline{\text{HDS2}}$ 信号はHPI-8のデータ転送を制御します。ただし、システムに応じてこれらの機能を入れ替えることができます。図4-3に、これらの入力の等価回路を示します。

図4-3. HPIストローブと選択ロジック

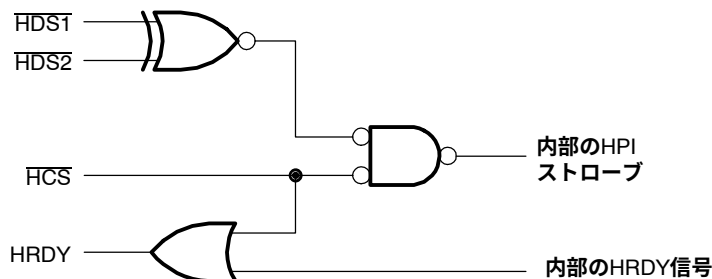


図4-3では、転送制御に使われる内部のHPIストローブ信号が、3つの入力信号( $\overline{HCS}$ 、 $\overline{HDS1}$ 、 $\overline{HDS2}$ )から生成されていることが示されています。なお、HPI-8のアクセス・サイクル制御に $\overline{HDS1}$ と $\overline{HDS2}$ ではなく $\overline{HCS}$ が使われている場合は、HRDYの動作に影響します( $\overline{HCS}$ はHRDYをイネーブルにし、 $\overline{HCS}$ がハイの場合はHRDYも強制的にハイになるため)。また、 $\overline{HDS1}$ と $\overline{HDS2}$ はXNOR処理されるため、これらの入力が全てローの場合はイネーブル状態になりません。

$\overline{HAS}$ 入力が使われない場合は必ず、HCNTL0/1、HBIL、HR/ $\overline{W}$ の各入力のサンプリングに内部ストローブの立ち下がりエッジが使われます。このため、 $\overline{HAS}$ が使われない場合は、 $\overline{HCS}$ 、 $\overline{HDS1}$ 、または $\overline{HDS2}$ のうち最も遅い信号が、HCNTL0/1、HBIL、HR/ $\overline{W}$ の各入力のサンプリングのタイミングを決定します。内部ストローブは制御入力をサンプリングするほか、HPI-8の1サイクルの区切りも定義します。内部ストローブのこの機能については、4-14ページの第4.4節「ホストからHPI-8へのリード/ライト・アクセス」を参照してください。

$\overline{HAS}$ 入力を使用して、HCNTL0/1、HBIL、HR/ $\overline{W}$ のサンプルをする場合、これらの信号をアクセス・サイクルの早い段階でサンプルできるので、アドレスからデータ情報にバス状態を切り替えるための時間をとることができます。この追加時間により、アドレスおよびデータ・バスを共用するホストとの接続が容易になります。この種のホストにはアドレス・ラッチ・イネーブル(ALE)信号が備わっていて、通常、 $\overline{HAS}$ 入力のドライブに使用できます。

2つの制御ピン(HCNTL0とHCNTL1)は、アクセス先となるHPI-8の内部レジスタを指定します。表4-4では、HCNTL0/1ピンの機能を示しています。



表4-4. HPI-8の入力制御信号とその機能

| HCNTL1 | HCNTL0 | 説明                                                                                               |
|--------|--------|--------------------------------------------------------------------------------------------------|
| 0      | 0      | ホストはHPI制御レジスタ(HPIC)からのリード、またはHPICへのライトを実行できます。                                                   |
| 0      | 1      | ホストはHPIデータ・ラッチに対してリード/ライトを実行できます。リードが1回行われるごとに、HPIAは自動的にポストインクリメントされ、ライトが1回行われるごとにプリインクリメントされます。 |
| 1      | 0      | ホストはアドレス・レジスタ(HPIA)に対してリード/ライトを実行できます。このレジスタは'C54xのオンチップRAM上のアドレスを指すポインタとして機能します。                |
| 1      | 1      | ホストはHPIデータ・ラッチに対してリード/ライトを実行できます。HPIAには影響ありません。                                                  |

#### 4.3.1 HPI-8のアドレス・レジスタとメモリー・マップ

ホストは'C54xのオンチップ・メモリーを指すポインタとしてHPIAレジスタを使用し、HPI-8を介してオンチップRAMの全域にアクセスすることができます。'C54xデバイスの内部メモリー・マップはそれぞれ異なっているので、HPI-8がアクセスできるアドレス範囲もデバイスによって異なります。たとえば、'C5410には、'C5402よりも多くのオンチップRAMが内蔵されています。図4-4に、'C5402デバイスと'C5410デバイスのHPI-8メモリー・マップを示します。

図4-4. HPI-8メモリー・マップ

| TMS320VC5402 |                           | TMS320VC5410 |                            |
|--------------|---------------------------|--------------|----------------------------|
| 0000h        | 予約                        | 0000h        | 予約                         |
| 005Fh        |                           | 005Fh        |                            |
| 0060h        | スクラッチ・パッド<br>RAM          | 0060h        | スクラッチ・パッド<br>RAM           |
| 007Fh        |                           | 007Fh        |                            |
| 0080h        | オンチップDARAM<br>(16K×16ビット) | 0080h        | オンチップDARAM                 |
| 3FFFh        |                           | 1FFFh        |                            |
| 4000h        | 未定義                       | 2000h        | オンチップSARAM1<br>(24K×16ビット) |
| FFFFh        |                           | 7FFFh        |                            |
|              |                           | 8000h        | 未定義                        |
|              |                           | 17FFFh       |                            |
|              |                           | 18000h       | オンチップSARAM2<br>(32K×16ビット) |
|              |                           | 1FFFFh       |                            |

オンチップRAMのすべてのブロック(プログラムRAMとデータRAM)は、HPIメモリー・マップ内で連続的なアドレス領域としてマッピングされています。このメモリー・マップ内のアドレスをユーザーがマッピングし直すことはできません(HPI-8メモリー・マップは、プログラム可能なレジスタ・ビットの影響を受けません)。

### 4.3.2 HPI-8の拡張アドレッシング

4

オンチップRAMの領域が64Kのアドレスを超えるデバイスに対応するために、HPI-8には拡張アドレッシング機能が組み込まれています。HPI-8では、オンチップRAMの拡張ページにアクセスするために、7つの拡張アドレス・ビットを使用します。ホストは、HPICレジスタの拡張HPIアドレス・ビット(XHPIA)を使ってこの拡張アドレスを変更できます。ホストがXHPIAビットを1にセットすると、拡張アドレス・ビットを表す7ビットのレジスタ(HPIA16:22)が、HPIAレジスタに代わってアクセス可能になります。これらの拡張アドレス・ビットを初期化するためには、ホストがHPIAレジスタへのライト・アクセスを実行し、各バイトの下位7ビットにHPIA16:22の値を設定する必要があります。このライト・アクセスに関しては、第1バイトと第2バイトの値が両方とも同じレジスタへライトされます。したがって、両方のバイトが同じ値でない場合は、第2バイトの値が拡張アドレスの初期設定に使われ、第1バイトの値は破棄されます。

拡張アドレス・ビットの初期化後、HPIAレジスタ内のHPIアドレス・ビット(下位16ビット)へのアクセスを復帰するためにホストはXHPIAビットをクリアする必要があります。また、自動インクリメント機能が正しく動作するためにも、XHPIAビットを0にクリアしなければなりません。XHPIAビットが1にセットされていると、自動インクリメント機能が正しく動作しません。

なお、XHPIAも7ビットの拡張アドレス・ビットもリセット後には初期化されないため、'C54xをリセットした後は必ず、ホストがこれらのビットを初期化しなければなりません。XHPIAビットの詳細は、4-12ページの第4.3.4節「HPI-8の制御レジスタ・ビットとその機能」で説明されています。

### 4.3.3 アドレスの自動インクリメント

HPI-8のアドレス自動インクリメント機能を使用すれば、オンチップRAM内の連続するアドレスのワードに容易にアクセスすることができます。自動インクリメント機能がイネーブルの場合は、1回アクセスするごとにHPIAレジスタ内の値が自動的にインクリメントされます。アクセス・タイムは変わりませんが、メモリー・アクセスを行うたびにホストがHPIAレジスタを更新する必要がないので、パフォーマンスが向上します。HCNTL0ピンを1、HCNTL1ピンを0にセットすると、自動インクリメント機能がイネーブルになります。なお、拡張オンチップRAMを搭載したデバイスでは、HPICレジスタのXHPIAビットを0にクリアしておかないと自動インクリメント機能は正しく動作しません。

4

自動インクリメントがイネーブルのときには、データのリードによりHPIAがポストインクリメントされ、データのライトではHPIAがプリインクリメントされます。このため、自動インクリメントを使って特定のアドレスにライトする場合は、HPIAレジスタをライトするアドレスに対し1を引いた値に設定する必要があります。自動インクリメント機能はHPIAの16ビットすべてに影響し、拡張オンチップRAMを搭載したデバイス('C5410を除く)では拡張アドレスにも影響します。HPIアドレスが0FFFFFFhに設定されていて、自動インクリメント機能がオンになっている場合は、次のアクセスでHPIアドレスが010000hに変更されます。'C5410の自動インクリメント回路はHPIの拡張アドレスに影響しないため、上記の例の場合、'C5410のHPIアドレスは000000hに変更されます。

## 4.3.4 HPI-8の制御レジスタ・ビットとその機能

HPICレジスタのビットは、HPI-8の動作の制御と状態の表示を行います。これらのビットには、BOB(第1バイトか第2バイトを上位として選択)、DSPINTとHINT('C54xとホストの割り込みをそれぞれ発生させるために使用)、XHPIA(ホストが拡張アドレスにアクセスするために使用)、HPIENA(HPI-8の状態(イネーブル/ディセーブル))があります。表4-5で、HPICの各ビットの機能が詳しく説明されています。

## 4

表4-5. HPI制御レジスタ(HPIC)のビット説明

| ビット    | リセット値 | 機能                                                                                                                                                                                                                                                      |
|--------|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| BOB    | 0     | バイト順ビット。このビットは、転送する2つのバイトに対して、どちらが上位になるかを決定します。BOB=1の場合は最初に転送されるバイト(第1バイト)が下位になり、BOB=0の場合は第1バイトが上位になります。このビットにアクセス(読み書き)できるのはホストのみで、データまたはアドレス・レジスタに最初にアクセスする前に必ず初期化しておく必要があります。                                                                        |
| DSPINT | 0     | ホストから'C54xへの割り込み。ホストがこのビットに1をライトすると、'C54xの割り込みが発生されます。このビットにライトできるのはホストだけで、ホストも'C54xもこのビットをリードすると常に0が得られます。ホストがHPICにライトを行うときには、両方のバイトで同じ値をライトする必要があります。DSPINTの詳細については、4-23ページの第4.5節を参照してください。                                                           |
| HINT   | 0     | 'C54xからホストへの割り込み。このビットは、ホストへの割り込みに使われる、'C54xのHINT出力の状態を決定します。HINTビットが1にセットされている場合はHINT出力がローにドライブされ、このビットが0にクリアされると出力はハイにドライブされます。HINTビットをセットできるのは'C54xだけですが、これをクリアできるのはホストに限られます。ホストがHINTビットをクリアする場合は、ビットに1をライトします。HINTの詳細については、4-23ページの第4.5節を参照してください。 |
| XHPIA  | X     | 拡張アドレス・イネーブル。XHPIA=1の場合は、ホストがHPIAレジスタにライトしたデータが上位ビットHPIA[n:16]にロードされます。XHPIA=0の場合は、ホストがHPIAにライトしたデータがHPIA[15:0]にロードされます。自動インクリメント・モードでは、n+1個のアドレス・ビットすべてに対してインクリメントが行われます(VC5410は下位16ビットのみ)。HPIAレジスタのリードも同じ方法で行われます。なお、このビットにアクセスできるのはホストに限られます。        |
| HPIENA | X     | HPIイネーブル・ステータス・ビット。このビットはHPIENAピンのリセット時の値をラッチします。また、'C54xはこのビットを使ってHPI-8がイネーブルかディセーブルかを判断することができます。このビットはライトの影響を受けません。また、ホストはこのビットを使用できません。<br>注記 このビットはどのデバイスでも使用できるとは限りません。詳細については、以降に記載するHPICレジスタの図を参照してください。                                        |

HPICは、ホスト側からは上位バイトと下位バイトが同じ内容の16ビット・レジスタとして、構成されます(ただし、前述で説明したように、いくつかのビットへのアクセスは制限されます)。HPICの上位8ビットは、'C54x側では使われません。ホストはHCNTL0/1を設定してHPICレジスタにアクセスし、8ビットのHPI-8データ・バスに対して連続した2バイトのアクセスを実行します。ホストがHPICにライトを行うときには、第1バイトと第2バイトの両方が同じ値でなければなりません。'C54x側では、HPICはデータ・メモリー空間内のメモリー・マップド・レジスタ(アドレス002Ch)に割り当てられています。

図4-5に、HPICの各ビットの配置を示します。図中で、リード処理に対して指定されている0は、そのビットが常に0でリードされることを示します。同様に、Xはこのビットのリード値が不定であることを示します。ライト処理に対して指定されている1は、ビットに1をライトする必要があることを示します。一方、ライト処理のXは、ビットに0か1をライトできることを示します。なお、未使用のビットは将来的な拡張のために予約されているため、特に指定がない限り、これらのビットには0をライトしてください。

図4-5. HPICレジスタ – ホストと'C54xによるアクセス

| ホストのHPICからのリード   |                    |      |                     |   |     |      |                    |      |        |   |     |
|------------------|--------------------|------|---------------------|---|-----|------|--------------------|------|--------|---|-----|
| 15-13            | 12                 | 11   | 10-9                |   | 8   | 7-5  | 4                  | 3    | 2      | 1 | 0   |
| 0                | XHPIA <sup>†</sup> | HINT | 0                   |   | BOB | 0    | XHPIA <sup>†</sup> | HINT | 0      | X | BOB |
| ホストのHPICへのライト    |                    |      |                     |   |     |      |                    |      |        |   |     |
| 15-13            | 12                 | 11   | 10                  | 9 | 8   | 7-5  | 4                  | 3    | 2      | 1 | 0   |
| X                | XHPIA <sup>†</sup> | HINT | DSPINT              | X | BOB | X    | XHPIA <sup>†</sup> | HINT | DSPINT | X | BOB |
| 'C54xのHPICからのリード |                    |      |                     |   |     |      |                    |      |        |   |     |
| 15-8             |                    |      | 7                   |   |     | 6-4  | 3                  | 2    | 1      | 0 |     |
| 0                |                    |      | HPIENA <sup>‡</sup> |   |     | 0    | HINT               | 0    | X      | 0 |     |
| 'C54xのHPICへのライト  |                    |      |                     |   |     |      |                    |      |        |   |     |
| 15-4             |                    |      |                     |   |     | 3    | 2                  | 1    | 0      |   |     |
| X                |                    |      |                     |   |     | HINT | X                  | 1    | X      |   |     |

Xは、ライトの影響を受けないビットか、1と0のどちらかでリードされるビットを示します。

<sup>†</sup> このビットは、オンチップRAMが拡張アドレスにマッピングされている'C54xデバイスでのみ使用できます。

<sup>‡</sup> このビットは、'C5410では使用できません。

'C54xでライト可能なHINTビットは、ホスト・インターフェイス側では2回リードします。したがって、ホストが2回のリードを行う間に'C54xがこのビットを変更した場合は、ホストがリードした第1バイトと第2バイトのデータが異なる可能性があります。表4-6に、ホストと'C54xによるHPICのリード/ライト・サイクルの特徴を示します。

表4-6. ホストと'C54xによるHPICのリード/ライトの特徴

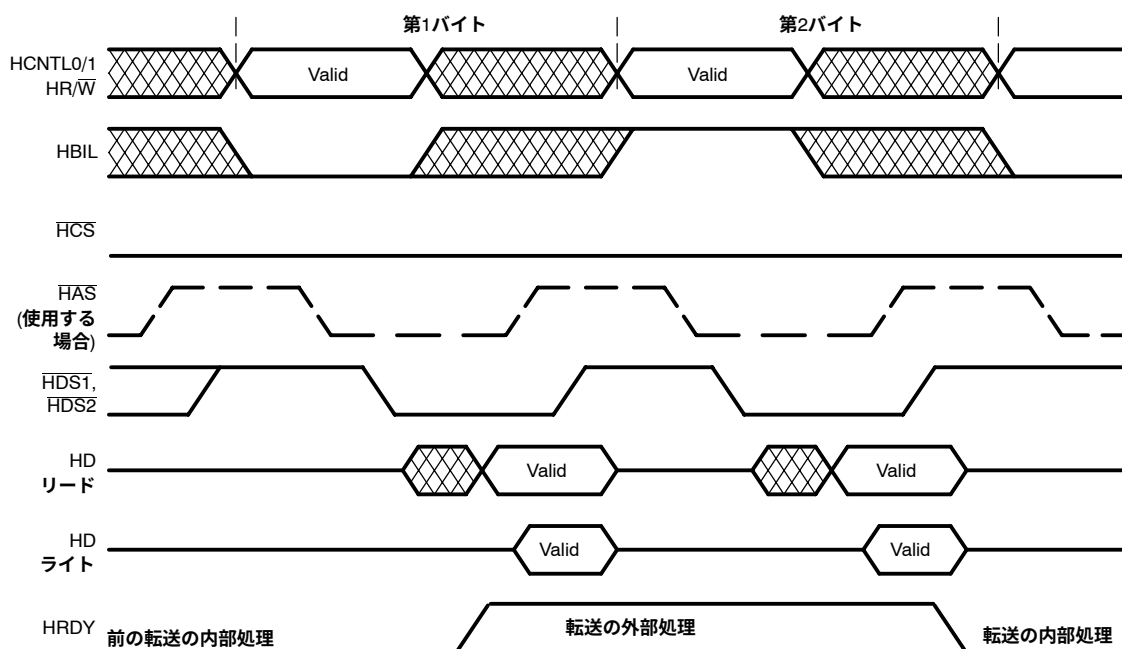
| デバイス  | リード   | ライト                      |
|-------|-------|--------------------------|
| ホスト   | 2バイト  | 2バイト(両方のバイトが同じでなければならない) |
| 'C54x | 16ビット | 16ビット                    |

### 4.4 ホストからHPI-8へのリード/ライト・アクセス

HPI-8の転送は、ホストがHPI-8の各レジスタとデータを交換する外部処理と、HPI-8ロジックがレジスタとオンチップRAM間のデータを交換する内部処理で構成されます。外部処理では、アクセスの種類(HPIA、HPIC、データ・アクセス)に関係なく常に2バイト単位の転送が必要です。転送の外部処理中には、ホストが第1バイトに対してHBIL入力をローにドライブし、次に第2バイトに対してHBIL入力をハイにドライブします。HPI-8転送の進行中、ホストがこの第1バイト/第2バイト(HBILロー/ハイ)のシーケンスを破ると、データが失われたり、予想外の動作が起こる可能性があります。このような状況から復帰するためには、ホストがバイトごとに正しいHBIL極性を使ってアクセスをやり直す必要があります。

また、ホストは1バイトごとにストローブ( $\overline{\text{HDS1}}$ 、 $\overline{\text{HDS2}}$ )と選択入力信号( $\overline{\text{HCS}}$ )を使ってアクセス時間を制御します。これらの入力から内部のHPIストローブを導出する方法については、4-6ページの第4.3節「HPI-8の動作について」を参照してください。HPIストローブの立ち下がりエッジは、第1バイトまたは第2バイトの転送開始を示します。このイベントは通常、ホストのバス・サイクルの始めに発生します。HPIストローブの立ち上がりエッジは、第1バイトまたは第2バイトの転送終了を示します。このイベントは通常、ホストのバス・サイクルの終わりに発生します。第2バイトの転送中に発生するHPIストローブの立ち上がりエッジは、サイクル全体の外部処理の終了を示し、これにより内部処理が開始されます。したがって、メモリー・リードの外部処理中にリードされるHPIDの値は、前のアクセスで指定されたアドレスの内容と同じになります。図4-6に、一般的なHPI-8のアクセスを示します。

図4-6. HPI-8のタイミング図



4

このサイクルは、ホストがHCNTL0/1、HR/W、HBIL、HCSをドライブし、転送の種類と、このサイクルがリードかライトかを指定するところから始まります。次にホストは、HAS信号(使用する場合)と、それに続いてデータ・ストロブ信号の1つを出力します。HRDYがまだハイになっていない場合は、前の内部処理が完了したらハイになり、データ転送が可能になります。HPI-8の外部サイクルが終了すると、HRDYがロー(HPI-8が非レディー状態)になり、転送の内部処理が行われている間はローのままになります。そして内部サイクルが完了すると、HRDYが再びハイに切り替わります。ただし、HCSがハイのときには、HRDYが常にハイの状態になることに注意してください。

### 4.4.1 HPI-8アクセスのレイテンシー

HPI-8のアクセス・タイムは、転送の外部処理と内部処理に必要な時間で決定されます。HPI-8で転送の外部処理に必要な時間は、各種信号ピンのセットアップ時間、ホールド時間、バッファ遅延時間などの一定の遅延時間です。通常、HPI-8転送の内部遅延はアクセス・タイムの大半を占め、 $\text{'C54x}$ のクロック周波数に依存します。この内部遅延は、アクセスの種類によって変わります。HPI-8のアクセス・タイムを正確に解析するには、内部遅延と外部遅延の両方を考慮に入れる必要があります。1回のアクセスの区切りはHPIストロープのエッジによって定義されるため、これらの信号についてタイミング要求を解析すれば十分です。HDS1、2のタイミング要求は各デバイスのデータ・シートに記載されています。なお、これらのタイミングには、HPI-8アクセスの外部遅延と内部遅延の両方が含まれています。以降に、HPI-8の内部アクセス遅延と $\text{'C54x}$ のクロック速度の関係について説明します。

最速のアクセスは次のとおりです。

- ☐ HPICレジスタとHPIAレジスタへのリード・アクセス
- ☐ HINTビットとDSPINTビットに0を書き込む、HPICレジスタへのライト・アクセス

これらのアクセスでは、ホストと内部レジスタ間でデータが交換されるだけなので、それ以上の内部アクセス・タイムはかかりません。外部処理アクセスが完了するとすぐに別のアクセスを開始できるため、これらのアクセスに対しては、HRDY信号がローにドライブされることはありません。

HINTビットかDSPINTビットに1をライトするHPICレジスタへのライト・アクセスの場合は、上述の場合よりもレイテンシーが長くなります。これらのアクセスでは、転送の外部処理が完了した後、HPI-8の内部割り込みロジックがその機能を実行するために多少時間がかかります。これらのアクセスの内部処理中には、 $\text{'C54x}$ のクロックで約3サイクルの間、HRDY信号がローになります。

HPI-8アクセスの中で最も遅く、最も頻繁に行われるものはメモリー・アクセスです。HPIAレジスタへのライト・アクセスや、HPIDレジスタへのリード/ライト・アクセスにより、内部メモリーの転送処理が開始され、HPIDと $\text{'C54x}$ のオンチップ・メモリー間で必要なデータが交換されます。HPI-8のメモリー・アクセスが発生するたびに、このような処理に $\text{'C54x}$ のクロックで数サイクル必要とします。HPI-8のメモリー転送に対して内部処理に必要な最小時間は $\text{'C54x}$ のクロックで5サイクルです。



HPI-8のメモリー転送の内部処理にかかる最大時間は、いくつかの要因によって決まります。HPI-8とダイレクト・メモリー・アクセス(DMA)コントローラは内部メモリーのアクセスに同じDMAバスを共用するため、1度にどちらか一方のモジュールしかバスにアクセスできません(DMAコントローラの詳細については、第3章を参照してください)。HPI-8とDMAコントローラの両方が同時にバスへアクセスしようとした場合は、内部のロジックがこの競合問題を解決します。HPI-8とDMAコントローラが同時にアクセスを開始した場合はHPI-8が優先され、DMAコントローラによる転送は5クロック・サイクル遅延されます。DMA転送の進行中にHPI-8がアクセスを開始した場合は、DMAによる現行ワードの転送が完了するまで、HPI-8のアクセスが遅延されます(HRDYはローになります)。この遅延時間中はホストがDMAコントローラを待ちますが、この時間は進行中のDMA転送の種類によって異なります。表4-7に、HPI-8のアクセス種類別による内部遅延を示します。

表4-7. アクセス種類別のHPI-8内部遅延

| アクセスの種類                                                |                              | 内部遅延                          |
|--------------------------------------------------------|------------------------------|-------------------------------|
| HPIA/HPICへのリード・アクセスか、HINTとDSPINTを0にクリアするHPICへのライト・アクセス |                              | 内部遅延なし                        |
| HINTかDSPINTを1にセットするHPICへのライト・アクセス                      |                              | 3 CPUクロック・サイクル                |
| DMAコントローラがインアクティブ時のメモリー・アクセス                           |                              | 5 CPUクロック・サイクル                |
| DMAコントローラがアクティブ時のメモリー・アクセス                             | DMAコントローラとオンチップRAM間の16ビット転送† | 5 CPUクロック・サイクル+4 CPUクロック・サイクル |
|                                                        | DMAコントローラとオンチップRAM間の32ビット転送† | 5 CPUクロック・サイクル+8 CPUクロック・サイクル |

† 注: DMAコントローラが外部メモリーにアクセスする場合は、ウェートステートによりレイテンシーが発生する可能性があります。

HPI-8のアクセス・タイムはアクセスの種類によって変わるため、ホストは常にHRDY出力をサンプルして、HPI-8の速度変化に合わせてバスのサイクル・タイムを調整しなければなりません。これを実行するために、HRDYをホストのREADY入力(使用可能な場合)に接続して、ホストに対してウェートステートを生成します。また、別の方法として、ホストのバス・サイクル・タイムをHPI-8の最も長いアクセス・タイムに合わせることも可能です。表4-8に、HPI-8のアクセスの種類別にホストでのHRDYによるウェートステートの発生状況について示します。

## ホストからHPI-8へのリード/ライト・アクセス

表4-8. ウエートステートの発生条件

| ウエートステートの発生(HRDYはローにドライブ) |     |                                          |
|---------------------------|-----|------------------------------------------|
| レジスタ                      | リード | ライト                                      |
| HPIC                      | なし  | DSPINT/HINTに1をライト – あり<br>その他の全サイクル – なし |
| HPIA                      | なし  | あり                                       |
| HPID                      | あり  | あり                                       |

HPI-8の内部アクセスにはプリフェッチ特性をもつため、特別な注意が必要となる場合があります。ランダム(不連続)転送中や、連続アクセスの間に多くの時間がかかる場合は、ホストが前のアクセスを終了してからリードを行うまでに、アクセス先の内容を'C54xが変更してしまう可能性があります。この場合は、リードされたデータと、実際のアクセス先のメモリー内容が異なる可能性があります。システム内でこの問題が起こりうる場合は、同じアドレスから2回リードを行うか、リード・アクセスの前にアドレスのライトを行うことで最新のデータを確実にリードすることができます。

#### 4.4.2 アクセス・シーケンスの例

データにアクセスする前に、ホストはまずHPIC、特にBOB(ビット0とビット8)を初期化してから、次にHPIAレジスタを初期化する必要があります。BOBの状態はHPIAレジスタのアクセスに影響するため、必ずこの順序で初期設定を行う必要があります。

拡張オンチップRAMを搭載したデバイスでは、ホストはHPIAレジスタにアクセスする前に、HPICのXHPIAビットも初期化する必要があります。XHPIAビットは、BOBを初期化するHPICへのライト・アクセスで同時に設定できます。XHPIAビットに1をセットすることで、ホストは7ビットのHPI拡張アドレスにアクセスできるようになります。その後ホストは、拡張アドレス(HPIA16:22)を指定する下位7ビットをHPIAレジスタにライトします。HPIの拡張アドレスを初期化するときには、第1バイトと第2バイトに同じ値をライトする必要があります。拡張アドレスが初期化されたら、ホストはもう1度HPICにアクセスしてXHPIAビットに0をライトし、HPIAレジスタの下位16アドレス・ビットへアクセスするように変更する必要があります。拡張アドレス機能の詳細については、4-10ページの第4.3.2節を参照してください。

BOBが初期化されると、ホストはHPIAレジスタ内の正しい位置にバイト列をライトできるようになります。ホストがHPIAにライトすると、C54xのオンチップ・メモリーが自動的にリードされ、所定のアドレスの内容が2つの8ビット・データ・ラッチ(HPIDレジスタの第1バイトと第2バイト)に転送されます。

表4-9には、HPI-8のメモリー・リードに使われる、HPICレジスタのBOBビットとXHPIAビットを初期化する際のシーケンスが示されています。なお、この表の6行目までは、拡張アドレスの初期化を示しています。この部分は拡張オンチップRAMを搭載したデバイスでのみ必要となります。この例では、BOBが0にクリアされ、FFFEhが格納されているオンチップ・メモリーの先頭アドレス(この場合は0060h番地)に対して、リードが要求されます。

## ホストからHPI-8へのリード/ライト・アクセス

表4-9. BOBとHPIAの初期化

| イベント                                      | HRDY | HD | HR/ $\overline{W}$ | HCNTL1/0 | HBIL | HPIC | HPIA | HPID |      |
|-------------------------------------------|------|----|--------------------|----------|------|------|------|------|------|
|                                           |      |    |                    |          |      |      |      | ラッチ1 | ラッチ2 |
| ホストによる<br>HPICのライト、<br>第1バイト <sup>†</sup> | 1    | 10 | 0                  | 00       | 0    | 10xx | xxxx | xxxx | xxxx |
| ホストによる<br>HPICのライト、<br>第2バイト <sup>†</sup> | 1    | 10 | 0                  | 00       | 1    | 1010 | xxxx | xxxx | xxxx |
| ホストによる<br>HPIAのライト、<br>第1バイト <sup>†</sup> | 1    | 00 | 0                  | 10       | 0    | 1010 | xxxx | xxxx | xxxx |
| ホストによる<br>HPIAのライト、<br>第2バイト <sup>†</sup> | 1    | 00 | 0                  | 10       | 1    | 1010 | xx00 | xxxx | xxxx |
| 内部遅延 <sup>†</sup>                         | 0    |    |                    |          |      |      |      | xxxx | xxxx |
| HPI RAMの内部<br>リード完了 <sup>†</sup>          | 1    |    |                    |          |      |      |      | xxxx | xxxx |
| ホストによる<br>HPICのライト、<br>第1バイト              | 1    | 00 | 0                  | 00       | 0    | 00xx | xxxx | xxxx | xxxx |
| ホストによる<br>HPICのライト、<br>第2バイト              | 1    | 00 | 0                  | 00       | 1    | 0000 | xxxx | xxxx | xxxx |
| ホストによる<br>HPIAのライト、<br>第1バイト              | 1    | 00 | 0                  | 10       | 0    | 0000 | 00xx | xxxx | xxxx |
| ホストによる<br>HPIAのライト、<br>第2バイト              | 1    | 60 | 0                  | 10       | 1    | 0000 | 0060 | xxxx | xxxx |
| 内部遅延                                      | 0    |    |                    |          |      |      | 0060 | xxxx | xxxx |
| RAMの内部リー<br>ド完了                           | 1    |    |                    |          |      |      | 0060 | FF   | FE   |

<sup>†</sup> これらのアクセスでは拡張アドレッシング機能が初期化されています。これらのアクセスは、拡張オンチップRAMを搭載したデバイスに対してのみ必要です。

表4-9の例では、HPICのBOBビットとXHPIAビットが先に初期化されます。次に、HPIAレジスタの下位7ビットにライト・アクセスすることにより、HPIの拡張アドレス・ビットが初期化されます。HPIの拡張アドレス・ビットを正しく初期化するには、このライト・アクセスの第1バイトと第2バイトで同じ値をライトする必要があります。なお、HPIAレジスタへのライト・アクセスによって、内部のリード・アクセスが自動的に開始されますが、アドレス・ビットの下位16ビットがまだ初期化されていないため、リードされるデータは不確定となります。ここまでの処理は、拡張オンチップRAMが搭載されたデバイスでのみ必要となります。

次にホストはXHPIAビットに0をライトし、HPIAレジスタに16ビットのアドレスをライトすることで、アドレス・ビットの下位16ビットを初期化します。前述と同様、HPIAレジスタへのライトによって、HPI-8の内部メモリー・アクセスが自動的に開始されます。この時点で、すべてのアドレス・ビットが初期設定されるので、内部リード処理により指定されたメモリーの内容が取り出されます。表4-9の最後の行は、RAMからの内部リード処理が完了した後のHPI-8の状態を示しています。これは、HPIAレジスタのライト終了後に遅延がおき、リードが完了してHPIDレジスタにデータが格納されている状態です。このデータを実際に取り出すには、ホストがさらにHPIDのリードを実行する必要があります。表4-10に、このアクセスに必要なシーケンスを示します。

表4-10. HPI-8へのリード・アクセス(自動インクリメントあり)

| イベント                | HRDY | HD | HR/W | HCNTL1/0 | HBIL | HPIC | HPIA | HPID |      |
|---------------------|------|----|------|----------|------|------|------|------|------|
|                     |      |    |      |          |      |      |      | ラッチ1 | ラッチ2 |
| ホストによるデータのリード、第1バイト | 1    | FF | 1    | 01       | 0    | 0000 | 0060 | FF   | FE   |
| ホストによるデータのリード、第2バイト | 1    | FE | 1    | 01       | 1    | 0000 | 0061 | FF   | FE   |
| 内部遅延                | 0    |    |      |          |      |      | 0061 | FF   | FE   |
| HPI RAMの内部リード完了     | 1    |    |      |          |      |      | 0061 | 6A   | BC   |

表4-10のアクセスでは、HPIDのリードにより得られるデータは、前のサイクル(表4-9に示すサイクル)で実行された処理によりHPIDに格納されたデータと同じになります。このHPIDリード・アクセス中に、HBILがローの場合は第1バイト、HBILがハイの場合は第2バイトのデータ・ラッチの内容がHDピンにドライブされます。表4-10で実行されるアクセスでは、さらに0061h番地のリードも開始されます(HCNTL1/0を01に設定することにより、このアクセスで自動インクリメントが指定されたため)。自動インクリメントを選択している場合は、16ビット・ワードが転送されるたびにインクリメントが行われます(各バイトごとではありません)。したがって、表4-10に示すようにHPIAが1つだけインクリメントされます。表4-10の最後の行は、第2バイトのリードに続いて遅延が生じた後、0061h番地の内容(6ABCh)がリードされてHPIDレジスタに格納されていることを示します。

HPIへのライト・アクセス中で、HBILピンがローの場合はホストからの着信データによって第1バイトのデータ・ラッチが上書きされ、HBILピンがハイの場合は第2バイトのデータ・ラッチが上書きされます。このライト・アクセスの最後には、両方のデータ・ラッチ内のバイトが16ビット・ワードとして、オンチップRAM内のHPIAレジスタで指定されたアドレスに転送されます。なお、自動インクリメントが選択されている場合は、メモリー・ライトの前にアドレスがインクリメントされます。

## ホストからHPI-8へのリード/ライト・アクセス

表4-11では、HPIのライト・アクセスの例が示されています。この例では、自動インクリメントがイネーブルになっているため、ライトが行われる前にHPIAレジスタがインクリメント(プリインクリメント)されます。前のリード・アクセスによってHPIAレジスタがポストインクリメントされたため、アドレス0061hはとばされます。ライトの内部処理が完了すると、オンチップRAMのアドレス0062hに1234hが格納されます。このライト後にアドレス0062hのリードが続く場合は、同じデータ(1234h)がリードされます。

表4-11. HPIへのライトアクセス(自動インクリメントあり)

| イベント                | HRDY | HD | HR/W | HCNTL1/0 | HBIL | HPIC | HPIA | HPID |      |
|---------------------|------|----|------|----------|------|------|------|------|------|
|                     |      |    |      |          |      |      |      | ラッチ1 | ラッチ2 |
| ホストによるデータのライト、第1バイト | 1    | 12 | 0    | 01       | 0    | 0000 | 0061 | 12   | FE   |
| ホストによるデータのライト、第2バイト | 1    | 34 | 0    | 01       | 1    | 0000 | 0062 | 12   | 34   |
| 内部遅延                | 0    |    |      |          |      |      | 0062 | 12   | 34   |
| RAMの内部ライト処理完了       | 1    |    |      |          |      |      | 0062 | 12   | 34   |

## 4.5 DSPINTとHINTの動作

ホストと'C54xは、HPICレジスタ内のビットを使って互いに割り込みを発生させることができます。以降の節で、この処理について説明します。

### 4.5.1 DSPINTを使用したホスト・デバイスによる'C54xへの割り込み

ホストがHPICのDSPINTビットに1をライトすると、'C54xに割り込みが発生します。この割り込みを使って、'C54xのIDLEモードを解除することができます。ホストと'C54xはこのビットを常に0としてリードします。なお、'C54xによるこのビットへのライトは無効です。ホストがDSPINTに1をライトしても、別の割り込みを発生する前に0をライトする必要はありません。また、このビットに0をライトしても無効になります。ホストがBOBまたはHINTにライトを行っている間は、DSPINTビットに1をライトしないようにしてください。これを実行すると、'C54xに対して不要な割り込みを発生してしまいます。

'C54x上では、ホストから'C54xへの割り込みベクタ・アドレスはxx64hになります。この割り込みは、IMR/IFRレジスタのビット9に割り当てられています。'C54xの割り込みベクタ・テーブルはオンチップRAMに再マッピングできるため、ホストはDSPINT割り込みベクタを初期化することで、事前にプログラムされている関数を実行するように'C54xに命令することができます。'C54xの割り込み・ベクタ・テーブルがオンチップRAMに再マッピングされると、ホストは割り込みを発生させる前に割り込みベクタ・テーブルに対して、次のことができますようになります。1) アドレスxx64hに分岐命令のオペコードをライトすること。2) アドレスxx65hに関数の開始アドレスをライトすること。このテクニックを使うときには、ホストが他の割り込みベクタを破壊しないよう注意する必要があります。

### 4.5.2 HINTを使用した'C54xによるホスト・デバイスへの割り込み

'C54xがHPICのHINTビットに1をライトすると、 $\overline{\text{HINT}}$ 出力がローにドライブされ、HINTビットは1としてリードされます。したがって、 $\overline{\text{HINT}}$ 信号はホスト・デバイスに対するアクティブ・ローの割り込み信号として使用できます。ホストはHINTビットに1をライトすることで、この割り込みをクリアし、 $\overline{\text{HINT}}$ 信号を解除することができます。このライトによってHINTビットがクリアされると、 $\overline{\text{HINT}}$ 信号がハイにドライブされます。'C54xまたはホストが0をライトしても、HINTビットは変わりません。

## 4.6 クロック・モード変更時のHPI-8転送に関する考慮事項

従来のスタンダード8ビットHPIとは異なり、HPI-8には非同期のホスト専用モード(HOM)が組み込まれていません。HPI-8の転送はすべて'C54xのクロックと同期されます。したがって、'C54xのクロック周波数が変わるとHPI-8のアクセス速度が変化するので、クロック・モードを変更するときには、HPI-8の転送に関して特別な配慮が必要となります。'C54xのクロック発生器のモードを変更する方法については、『TMS320C54x DSP Reference Set, Volume 1: CPU and Peripherals』(文書番号SPRU131)を参照してください。'C54xのクロック速度とHPI-8のアクセス速度の関係については、4-16ページの第4.4.1節を参照してください。

'C54xのクロック・モードがPLLモードからDIVモードに変わると、クロック速度は遅くなり、HPI-8のアクセス速度も遅くなります。この場合、ホストはHRDY出力を使ってウェイトステートを発生し、アクセス速度を調整することができます。図4-7に、クロック・モードを1通倍(MULT1)から2分周(DIV2)に変更したときのHPI-8のアクセス例を示します。

図4-7. DIVクロック・モードに切り替えた場合のHPI-8転送

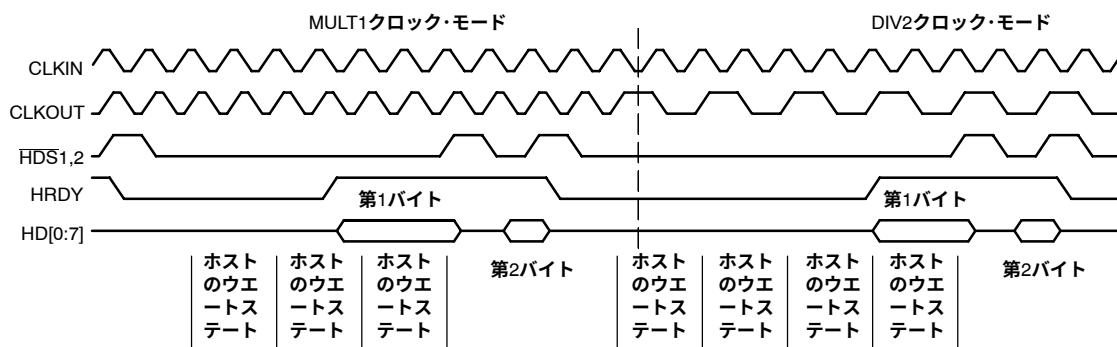


図4-7の例では、クロック・モード切り替えの前後で'C54xのクロック周波数は大きく変わっていないため、結果的にHPI-8のアクセス・タイムに与える影響も小さくなります。この場合、ホストはウェイトステートを1つ追加して、アクセス速度の変化に対処することができます。高いクロック通倍率が使われる場合には、分周モードに切り替えるとHPI-8のアクセス速度が大幅に低下するため、ホストのウェイトステートをさらに追加する必要があります。



## クロック・モード変更時のHPI-8転送に関する考慮事項

'C54xのクロック発生器が分周モードから逡倍モードに切り替わると、結果的に'C54xのクロック速度とHPI-8のアクセス速度が上がります。この場合、HRDYを使ってホストにウェートステートを挿入しているなら、ホストは図4-7と同様の方法でウェートステートの数を自動的に減らします。HPI-8のアクセス速度がホストのバス・サイクルよりも速くなる場合は、ウェートステートは不要になり、ホストのバス・サイクル・タイムによって新しいアクセス速度が定まります。

## 4.7 IDLE使用時の考慮事項

HPI-8では、'C54xがIDLEモードになっている間にホストがオンチップRAMにアクセスすることができます。これらのモード時には、HPI-8のアクセスに関する唯一の条件は、'C54xの入力クロック(CLKIN)がアクティブであることです。'C54xのIDLEモードの詳細については、『TMS320C54x DSP Reference Set, Volume 1: CPU and Peripherals』(文書番号SPRU131)を参照してください。

### 4

#### 4.7.1 IDLE1およびIDLE2モードでのHPI-8アクセス

IDLE1モードとIDLE2モードは'C54xのクロックに影響しないため、これらのモード時にはHPI-8転送に関する特別な配慮はいりません。しかし、これらのパワーダウン・モードに入る前にPLL回路をディセーブルにして、さらに消費電力を減らしたい場合もあります。このような場合にはHPI-8のアクセス速度に影響が出るため、特別な配慮が必要となります。クロック・モード変更時のHPI-8の使用法に関する考慮事項については、4-24ページの第4.6節を参照してください。

#### 4.7.2 IDLE3モードでのHPI-8アクセス

通常、HPI-8のアクセス速度はIDLE3パワーダウン・モードによる影響を受けます。これは、このモードが'C54xの内部クロック発生回路に影響するためです。'C54xがIDLE3モードに入ると、オンチップ・クロック発生器がディセーブルになり、HPI-8のアクセスが入力クロック・ソース(CLKIN)と自動的に同期されます。'C54xがIDLE3モードになっている間、外部から'C54xのCLKINピンにクロックが入力されている限りは、ホストはHPI-8にアクセスし続けることができます。HPI-8へのアクセスは、'C54xがIDLE3モードから抜けるまでCLKINと同期され続けます。割り込みによって'C54xがIDLE3モードから抜けると、HPI-8へのアクセスは自動的に'C54xのクロック発生器出力と同期されます。クロック発生器がPLLモードに設定されている場合は、PLLロック・タイマのカウントダウンが終わるまで、HPI-8へのアクセスが入力ソースと同期され続けます。'C54xのクロック生成器とプログラマブル・ロック・タイマの詳細については、『TMS320C54x DSP Reference Set, Volume 1: CPU and Peripherals』(文書番号SPRU131)を参照してください。

クロック通倍率として1が選択されている時に'C54xがIDLE3モードに入った場合、このモード中はHPI-8のアクセス速度は変わりません( $\times 1$ )。また、クロック生成器がDIV(分周)モードに設定されていて、'C54xがIDLE3モードに入った場合はHPI-8のアクセス速度が上がります( $\times 1$ )。どちらの場合も、IDLE3モード中はHPI-8のアクセス速度が下がることがないため、特別な配慮はいりません。

2 4 1よりも大きいクロック通倍率が選択されている時に'C54xがIDLE3モードに入った場合は、HPI-8のアクセス速度が下がります。この場合、ホストに対してHRDY出力を使って追加のウェイトステートを挿入し、アクセス速度の変化に対処することができます。図4-8に、この場合の例を示します。

図4-8. 'C54xがIDLE3モードの場合のHPI-8の転送

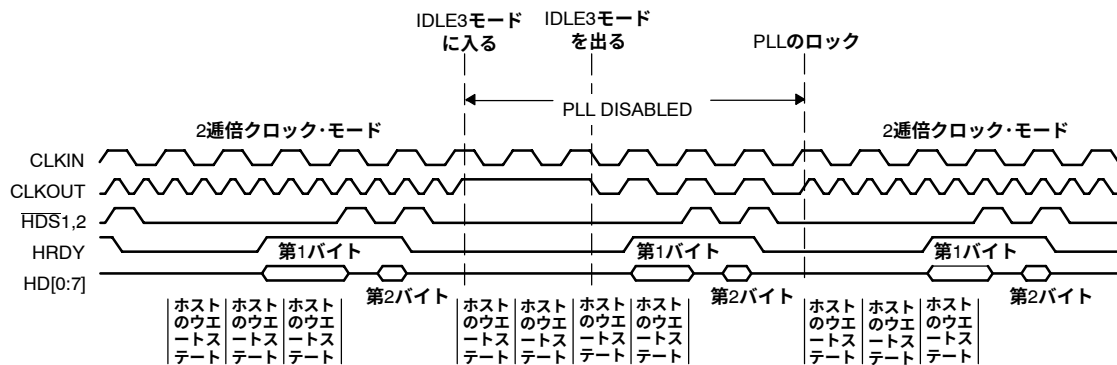


図4-8では、クロック発生器が通倍率2でPLLモードに設定されている状態で'C54xがIDLE3モードに入っています。HPI-8のアクセスは自動的に入力クロック・ソースと同期されるため、ホストはHRDYを用いて、メモリー・アクセス用のウェイトステートを追加挿入する必要があります。またこの図では、'C54xがIDLE3モードから出てクロック発生器が安定した後で、HPI-8のアクセスが通倍されたクロックと自動的に再同期されていることを示しています。

### 4.8 リセット時におけるHPI-8の動作

'C54xデバイスがリセットされると、HPI-8の動作に影響します。この節では、リセットがHPI-8の動作に与える影響について説明します。

#### 4.8.1 リセット後のHPI-8へのアクセス

4

'C54xデバイスのリセット中は、ホストからHPI-8にアクセスすることができません(ただし'C5410は例外です。詳細については第4.8.2節を参照してください)。'C54xデバイスのリセットが解除されると、ホストはHPI-8にアクセスできるようになります。HPI-8のアクセス速度は'C54xのクロック速度に左右されます。それに加え、リセットにより'C54xのクロック速度が再初期化されるので、ホストのアクセス速度とリセット後に選択されるHPI-8のアクセス速度が合うよう注意する必要があります。リセット時には、'C54xのクロック発生器が、クロック・モード・ピン(CLKMD1~3)で選択されたモードに初期化されます。ホストが'C54xのリセット・ピンとクロック・モード・ピンを制御している場合は、リセット後にホストがHPI-8のアクセス速度を設定できます。クロック発生器とクロック・モード・ピンの詳細については、『TMS320C54x DSP Reference Set, Volume 1: CPU and Peripherals』(文書番号SPRU131)を参照してください。

#### 4.8.2 リセット時のHPI-8へのアクセス('C5410のみ)

'C5410のHPI-8は、旧バージョンの8ビットHPIとの互換性を考慮して、リセット時のホスト専用モード(HOM)をサポートしています。リセット中には'C5410は動作しませんが、ホストはHPI-8にアクセスして、オンチップ・メモリーにプログラムやデータをダウンロードすることができます。リセット中のアクセスは'C54xのクロックと内部で同期されず、またHRDY出力はアクティブのままです(ただし、HPICのDSPINTビットかHINTビットに1がライトされている場合は例外です)。'C5410のリセット中にホストがHPICレジスタのDSPINTビットかHINTビットに1をライトすると、HPI-8が正しく機能しなくなり、HRDY出力がインアクティブになる可能性があるため、このライトは避けてください。

リセット中にHOMが使われている場合は、通常ホストが'C5410のリセット入力を制御するのが便利です。表4-12に、'C5410をリセットして、この間にオンチップRAMにプログラムをダウンロードする際のイベント・シーケンスを示します。

表4-12. リセット時のHPI-8の動作('C5410のみ)

| ホスト                     | 'C5410  | モード | 'C5410のクロック |
|-------------------------|---------|-----|-------------|
| 'C5410のクロックで6サイクル待機     | 動作中     | X   | 動作中         |
| RESETをローにして4クロック待機      | リセットに入る | HOM | 動作中         |
| 'C5410のクロック停止可能         | リセット中   | HOM | 停止または動作中    |
| オンチップRAMにプログラム/データをライト  | リセット中   | HOM | 停止または動作中    |
| DSPクロックが停止している場合は動作させる† | リセット中   | HOM | 動作中         |
| RESETをハイにする             | リセット中   | HOM | 動作中         |
| 'C5410のクロックで20サイクル待機    | リセット解除  | SAM | 動作中         |
| HPI-8にアクセス可能            | 動作中     | SAM | 動作中         |

† 'C5410のオンチップPLLを使用する場合は、十分なウェイクアップ時間を取る必要があります。

最初に、ホストは'C5410のリセット信号をローにドライブを始める6サイクル('C5410のクロックで)以上前に、HPI-8へのアクセスをストップします。ホストが'C5410のリセット信号をローにドライブしたら、少なくとも'C5410で4クロックの待機後にHPI-8にアクセスできるようになります。リセット時には、HPI-8のモードが自動的にHOMになるため、プログラムを高速にダウンロードすることができます。このとき、'C5410のクロックを停止することは可能ですが、'C5410が正しくリセットされるためにはリセット信号の立ち下がり時と立ち上がり時にクロックが動作していなければなりません。

ホストはオンチップRAMへのダウンロードを完了すると、HPI-8へのアクセスを終了し、'C5410のリセット信号をハイにドライブします。リセット信号の立ち上がりエッジ後、ホストが再度HPI-8にアクセスできるようになるためには、少なくとも20サイクルかかります。このサイクル数は、'C5410の内部リセット遅延によるものです。リセットが解除されると、HPI-8のモードが自動的にSAMになります。また、'C5410のHPI-8ブート・モードを使って、アドレス02000hからプログラムの実行を開始することができます。

#### 4.9 HPI-8データ・ピンの汎用I/Oピンとしての使用('C5410では使用不可)

HPI-8の8ビット双方向データ・バスは、汎用入出力(GPIO)ピンとして使用できます。この機能を使用できるのは、HPI-8がディセーブルになっているとき、つまりリセット中にHPIENAピンがローにドライブされているときに限られます。HPI-8データ・ピンのGPIO機能の制御には、2つのメモリー・マップド・レジスタ、汎用I/O制御レジスタ(GPIOCR)と汎用I/Oステータス・レジスタ(GPIOSR)が使われます。図4-9にGPIOCRのビット配置が、表4-13にこのレジスタの各ビットについての説明が示されています。

HD0～HD7を入力または出力として設定するには、GPIOCR内の方向ビット(DIRx)が使われます。2つのタイマが組み込まれたデバイスでは、タイマ1出力ビット(TOUT1)を使って、HINTピンにタイマ1を出力することができます。HPI-8がイネーブルになっている場合は、TOUT1ビットとDIRxビットが強制的に0にクリアされるため、汎用I/O機能は入力モードのみになります。

図4-9. 汎用I/O制御レジスタ(GPIOCR):MMRアドレス003Ch

|        |  |       |  |       |  |       |  |       |  |       |  |       |  |       |  |
|--------|--|-------|--|-------|--|-------|--|-------|--|-------|--|-------|--|-------|--|
| 15     |  | 14    |  |       |  |       |  | 8     |  |       |  |       |  |       |  |
| TOUT1† |  | 予約    |  |       |  |       |  |       |  |       |  |       |  |       |  |
| R/W-0  |  |       |  |       |  |       |  |       |  |       |  |       |  |       |  |
| 7      |  | 6     |  | 5     |  | 4     |  | 3     |  | 2     |  | 1     |  | 0     |  |
| DIR7   |  | DIR6  |  | DIR5  |  | DIR4  |  | DIR3  |  | DIR2  |  | DIR1  |  | DIR0  |  |
| R/W-0  |  | R/W-0 |  | R/W-0 |  | R/W-0 |  | R/W-0 |  | R/W-0 |  | R/W-0 |  | R/W-0 |  |

† オンチップ・タイマを2つ搭載したデバイスでのみ使用可能

注: R=リード、W=ライト

表4-13. 汎用I/O制御レジスタ(GPIOCR)のビット機能

| 番号   | 名前    | リセット値 | 機能                                                                                                                                                                                                                   |
|------|-------|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 15   | TOUT1 | 0     | タイマ1出力イネーブル・ビット。TOUT1ビットは、タイマ1出力のHINTピンへのドライブをイネーブル/ディセーブルにします。タイマ1出力は、HPI-8がディセーブルの場合にのみ可能です。なお、タイマが1つしかないデバイス上では、このビットは予約されています。<br><br>TOUT1 = 0    タイマ1出力は外部にドライブされません。<br>TOUT1 = 1    タイマ1出力がHINTピン上でドライブされます。 |
| 14～8 | 予約    | 0     | これらのピンは予約されているため、ライトの影響を受けません。                                                                                                                                                                                       |
| 7    | DIR7  | 0     | I/Oピン7の方向ビット。DIR7によって、HD7ピンが入力または出力として設定されます。<br><br>DIR7 = 0    HD7ピンが入力として設定されます。                                                                                                                                  |

表4-13. 汎用I/O制御レジスタ(GPIOCR)のビット機能 (続き)

| 番号 | 名前   | リセット<br>値 | 機能                                                                             |
|----|------|-----------|--------------------------------------------------------------------------------|
| 6  | DIR6 | 0         | DIR7 = 1      HD7ピンが出力として設定されます。HPI-8がイネーブルの場合はこのビットが強制的に0にされるため、ライトの影響を受けません。 |
|    |      |           | DIR6 = 0      HD6ピンが入力として設定されます。                                               |
|    |      |           | DIR6 = 1      HD6ピンが出力として設定されます。HPI-8がイネーブルの場合はこのビットが強制的に0にされるため、ライトの影響を受けません。 |
| 5  | DIR5 | 0         | I/Oピン5の方向ビット。DIR5によって、HD5ピンが入力または出力として設定されます。                                  |
|    |      |           | DIR5 = 0      HD5ピンが入力として設定されます。                                               |
|    |      |           | DIR5 = 1      HD5ピンが出力として設定されます。HPI-8がイネーブルの場合はこのビットが強制的に0にされるため、ライトの影響を受けません。 |
| 4  | DIR4 | 0         | I/Oピン4の方向ビット。DIR4によって、HD4ピンが入力または出力として設定されます。                                  |
|    |      |           | DIR4 = 0      HD4ピンが入力として設定されます。                                               |
|    |      |           | DIR4 = 1      HD4ピンが出力として設定されます。HPI-8がイネーブルの場合はこのビットが強制的に0にされるため、ライトの影響を受けません。 |
| 3  | DIR3 | 0         | I/Oピン3の方向ビット。DIR3によって、HD3ピンが入力または出力として設定されます。                                  |
|    |      |           | DIR3 = 0      HD3ピンが入力として設定されます。                                               |
|    |      |           | DIR3 = 1      HD3ピンが出力として設定されます。HPI-8がイネーブルの場合はこのビットが強制的に0にされるため、ライトの影響を受けません。 |
| 2  | DIR2 | 0         | I/Oピン2の方向ビット。DIR2によって、HD2ピンが入力または出力として設定されます。                                  |
|    |      |           | DIR2 = 0      HD2ピンが入力として設定されます。                                               |
|    |      |           | DIR2 = 1      HD2ピンが出力として設定されます。HPI-8がイネーブルの場合はこのビットが強制的に0にされるため、ライトの影響を受けません。 |
| 1  | DIR1 | 0         | I/Oピン1の方向ビット。DIR1によって、HD1ピンが入力または出力として設定されます。                                  |
|    |      |           | DIR1 = 0      HD1ピンが入力として設定されます。                                               |
|    |      |           | DIR1 = 1      HD1ピンが出力として設定されます。HPI-8がイネーブルの場合はこのビットが強制的に0にされるため、ライトの影響を受けません。 |
| 0  | DIR0 | 0         | I/Oピン0の方向ビット。DIR0によって、HD0ピンが入力または出力として設定されます。                                  |

表4-13. 汎用I/O制御レジスタ(GPIOCR)のビット機能 (続き)

| 番号 | 名前 | リセット<br>値 | 機能                                                               |
|----|----|-----------|------------------------------------------------------------------|
|    |    | DIR0 = 0  | HD0ピンが入力として設定されます。                                               |
|    |    | DIR0 = 1  | HD0ピンが出力として設定されます。HPI-8がイネーブルの場合はこのビットが強制的に0にされるため、ライトの影響を受けません。 |

4

GPIOピン(HD<sub>x</sub>, x=0:7)の状態をモニタするには、汎用I/Oステータス・レジスタ(GPIOISR)のビットが使われます。HD<sub>x</sub>ピンが入力として設定されている場合(GPIOISRのDIR<sub>x</sub>ビットに0をライト)は、GPIOISR内の対応ビットをリードすることで、そのピンのロジック・レベルが得られます。同様に、HD<sub>x</sub>ピンが出力として設定されている場合は、GPIOISR内のビットにライトされた値が対応するピンにドライブされます。図4-10にGPIOISRのビット配置が、表4-14にこのレジスタの各ビットについて説明が示されています。

図4-10. 汎用I/Oステータス・レジスタ(GPIOISR):MMRアドレス003Dh

|    |   |
|----|---|
| 15 | 8 |
| 予約 |   |

|       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|
| 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
| IO7   | IO6   | IO5   | IO4   | IO3   | IO2   | IO1   | IO0   |
| R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |

注: R=リード, W=ライト

† HDピンが入力として設定されている場合は、IO<sub>x</sub>ビットへのライトは無効です。

表4-14. 汎用I/Oステータス・レジスタ(GPIOISR)のビット機能

| 番号   | 名前  | リセット値 | 機能                                                                                                                                                                                                                                                                        |
|------|-----|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 15-8 | 予約  | 0     | これらのピンは予約されているため、ライトの影響を受けません。                                                                                                                                                                                                                                            |
| 7    | IO7 | X     | IO7 - I/Oピン7の状態ビット。このビットは、HD7ピンのロジック・レベルを反映します。HD7ピンが入力(GPIOICRのDIR7=0)として設定されている場合は、IO7ビットはピンのロジック・レベル(1または0)をラッチします。HD7ピンが出力(GPIOICRのDIR7=1)として設定されている場合は、IO7ビットにライトされたロジック・レベル(1または0)にHD7ピンがドライブされます。<br><br>IO7 = 0      HD7の入力が外部でローにドライブされるか、HD7の出力が内部でローにドライブされます。 |



表4-14. 汎用I/Oステータス・レジスタ(GPIOSR)のビット機能 (続き)

| 番号 | 名前 | リセット値   | 機能                                           |
|----|----|---------|----------------------------------------------|
|    |    | IO7 = 1 | HD7の入力が外部でハイにドライブされるか、HD7の出力が内部でハイにドライブされます。 |

表4-14. 汎用I/Oステータス・レジスタ(GPIOISR)のビット機能 (続き)

| 番号 | 名前  | リセット値 | 機能                                                                                                                                                                                                                                            |
|----|-----|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 6  | IO6 | X     | IO6 – I/Oピン6の状態ビット。このビットは、HD6ピンのロジック・レベルを反映します。HD6ピンが入力(GPIOCRのDIR6=0)として設定されている場合は、IO6ビットはピンのロジック・レベル(1または0)をラッチします。HD6ピンが出力として設定されている場合は、IO6ビットへのライトは無効です。HD6ピンが出力(GPIOCRのDIR6=1)として設定されている場合は、IO6ビットにライトされたロジック・レベル(1または0)にHD6ピンがドライブされます。 |
|    |     |       | IO6 = 0      HD6の入力が外部でローにドライブされるか、HD6の出力が内部でローにドライブされます。                                                                                                                                                                                     |
|    |     |       | IO6 = 1      HD6の入力が外部でハイにドライブされるか、HD6の出力が内部でハイにドライブされます。                                                                                                                                                                                     |
| 5  | IO5 | X     | IO5 – I/Oピン5の状態ビット。このビットは、HD5ピンのロジック・レベルを反映します。HD5ピンが入力(GPIOCRのDIR5=0)として設定されている場合は、IO5ビットはピンのロジック・レベル(1または0)をラッチします。HD5ピンが出力として設定されている場合は、IO5ビットへのライトは無効です。HD5ピンが出力(GPIOCRのDIR5=1)として設定されている場合は、IO5ビットにライトされたロジック・レベル(1または0)にHD5ピンがドライブされます。 |
|    |     |       | IO5 = 0      HD5の入力が外部でローにドライブされるか、HD5の出力が内部でローにドライブされます。                                                                                                                                                                                     |
|    |     |       | IO5 = 1      HD5の入力が外部でハイにドライブされるか、HD5の出力が内部でハイにドライブされます。                                                                                                                                                                                     |
| 4  | IO4 | X     | IO4 – I/Oピン4の状態ビット。このビットは、HD4ピンのロジック・レベルを反映します。HD4ピンが入力(GPIOCRのDIR4=0)として設定されている場合は、IO4ビットはピンのロジック・レベル(1または0)をラッチします。HD4ピンが出力として設定されている場合は、IO4ビットへのライトは無効です。HD4ピンが出力(GPIOCRのDIR4=1)として設定されている場合は、IO4ビットにライトされたロジック・レベル(1または0)にHD4ピンがドライブされます。 |
|    |     |       | IO4 = 0      HD4の入力が外部でローにドライブされるか、HD4の出力が内部でローにドライブされます。                                                                                                                                                                                     |
|    |     |       | IO4 = 1      HD4の入力が外部でハイにドライブされるか、HD4の出力が内部でハイにドライブされます。                                                                                                                                                                                     |

表4-14. 汎用I/Oステータス・レジスタ(GPIOISR)のビット機能 (続き)

| 番号 | 名前  | リセット値 | 機能                                                                                                                                                                                                                                                                                                                                                                                     |
|----|-----|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3  | IO3 | X     | <p>IO3 – I/Oピン3の状態ビット。このビットは、HD3ピンのロジック・レベルを反映します。HD3ピンが入力(GPIOCRのDIR3=0)として設定されている場合は、IO3ビットはピンのロジック・レベル(1または0)をラッチします。HD3ピンが入力として設定されている場合は、IO3ビットへのライトは無効です。HD3ピンが出力(GPIOCRのDIR3=1)として設定されている場合は、IO3ビットにライトされたロジック・レベル(1または0)にHD3ピンがドライブされます。</p> <p>IO3 = 0      HD3の入力が外部でローにドライブされるか、HD3の出力が内部でローにドライブされます。</p> <p>IO3 = 1      HD3の入力が外部でハイにドライブされるか、HD3の出力が内部でハイにドライブされます。</p> |
| 2  | IO2 | X     | <p>IO2 – I/Oピン2の状態ビット。このビットは、HD2ピンのロジック・レベルを反映します。HD2ピンが入力(GPIOCRのDIR2=0)として設定されている場合は、IO2ビットはピンのロジック・レベル(1または0)をラッチします。HD2ピンが入力として設定されている場合は、IO2ビットへのライトは無効です。HD2ピンが出力(GPIOCRのDIR2=1)として設定されている場合は、IO2ビットにライトされたロジック・レベル(1または0)にHD2ピンがドライブされます。</p> <p>IO2 = 0      HD2の入力が外部でローにドライブされるか、HD2の出力が内部でローにドライブされます。</p> <p>IO2 = 1      HD2の入力が外部でハイにドライブされるか、HD2の出力が内部でハイにドライブされます。</p> |
| 1  | IO1 | X     | <p>IO1 – I/Oピン1の状態ビット。このビットは、HD1ピンのロジック・レベルを反映します。HD1ピンが入力(GPIOCRのDIR1=0)として設定されている場合は、IO1ビットはピンのロジック・レベル(1または0)をラッチします。HD1ピンが入力として設定されている場合は、IO1ビットへのライトは無効です。HD1ピンが出力(GPIOCRのDIR1=1)として設定されている場合は、IO1ビットにライトされたロジック・レベル(1または0)にHD1ピンがドライブされます。</p> <p>IO1 = 0      HD1の入力が外部でローにドライブされるか、HD1の出力が内部でローにドライブされます。</p> <p>IO1 = 1      HD1の入力が外部でハイにドライブされるか、HD1の出力が内部でハイにドライブされます。</p> |

表4-14. 汎用I/Oステータス・レジスタ(GPIOISR)のビット機能 (続き)

| 番号 | 名前  | リセット値   | 機能                                                                                                                                                                                                                                              |
|----|-----|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0  | IO0 | X       | IO0 – I/Oピン0の状態ビット。このビットは、HD0ピンのロジック・レベルを反映します。HD0ピンが入力(GPIOICRのDIR0=0)として設定されている場合は、IO0ビットはピンのロジック・レベル(1または0)をラッチします。HD0ピンが入力として設定されている場合は、IO0ビットへのライトは無効です。HD0ピンが出力(GPIOICRのDIR0=1)として設定されている場合は、IO0ビットにライトされたロジック・レベル(1または0)にHD0ピンがドライブされます。 |
|    |     | IO0 = 0 | HD0の入力が外部でローにドライブされるか、HD0の出力が内部でローにドライブされます。                                                                                                                                                                                                    |
|    |     | IO0 = 1 | HD0の入力が外部でハイにドライブされるか、HD0の出力が内部でハイにドライブされます。                                                                                                                                                                                                    |

#### 4.9.1 GPIO機能の使用法

HPIデータ・ピンを汎用入出力として使用するには、まず各ピンを適切に設定する必要があります。設定が終了したら、GPIOISRのリードとライトによって各ピンをモニタまたは操作することができます。図4-11に、HPI-8のGPIO機能を使用する際のプログラム例を示します。

図4-11. GPIOコードの例

```

GPIOCR    .set    3Ch    ;MMR address for GPIOCR is 3Ch
GPIOISR    .set    3Dh    ;MMR address for GPIOISR is 3Dh

.text
STM        #0F0h, GPIOCR ;Configure HD0-3 as in, and
                                ;HD4-7 as out.
    .      .      .

LDM        GPIOISR, A      ;Get GPIOISR value.
AND        #0Fh, A         ;Mask off MSBs.
STLM       A, AR3          ;Store value of HD0-3 in AR3.
STM        #050h, GPIOISR ;Set HD4-7 to 0101b.
    .      .      .

```

この例では、HPIデータ・バスの下位4ビット(HD0～3)が汎用入力として設定され、上位4ビット(HD4～7)が出力として設定されています。HD0～3の状態がリードされてAR3に格納された後、上位4ビット(HD4～7)が0101bにセットされます。

# 拡張16ビット・ホスト・ポート・インターフェイス (HPI-16)

拡張16ビット・ホスト・ポート・インターフェイス(HPI-16)は、各種ホスト・プロセッサと'C54xとのインターフェイスをとるために設計されたHPIの改良バージョンです。

| Topic                                 | Page |
|---------------------------------------|------|
| 5.1 HPI-16の動作の概要 .....                | 5-2  |
| 5.2 共用モード .....                       | 5-7  |
| 5.3 分離モード .....                       | 5-15 |
| 5.4 HPI-16のメモリー・マップ .....             | 5-18 |
| 5.5 HPI-16とDMAの相互作用 .....             | 5-19 |
| 5.6 リセット時のHPI-16の動作 .....             | 5-21 |
| 5.7 IDLEn時のHPI-16の動作 .....            | 5-21 |
| 5.8 DSPクロック・モードの変更によるHPI-16への影響 ..... | 5-22 |

## 5.1 HPI-16の動作の概要

HPIは、パラレル・バス・インターフェイスを用いて、ホスト・プロセッサによる内部DSPメモリーへのアクセスを実現します。HPIはホストとDSPの間のデータ転送に使用され、データのバッファ、リアルタイム・データ・ログの収集、およびメッセージ通信処理を可能にします。

テキサス・インスツルメンツが提供する'C54x用のHPIには、いくつかの種類があります。現在のところ、8ビットHPIは16ビット・データをサポートしますが、ホスト・プロセッサとの接続は8ビットの双方向バスで行います。16ビット・データ・ワードを転送するために、上位バイトと下位バイトのバイト識別を行い、連続した2回のバイト転送が必要です。

HPI-16は、'C54xの8ビット・ホスト・ポート・インターフェイスを16ビットに拡張したバージョンです。8ビットHPIと同様、スレーブとして機能し、ホスト・プロセッサがDSPのCPUの介入なしで内部メモリーにアクセスできます。16ビットの双方向データ・バスによりバイト識別を必要としません。さらに、1ワードのアクセスを完了するために必要なホスト転送は1回だけです。このHPI-16により、多様なホスト・プロセッサとTMS320 DSPを接続することができます。

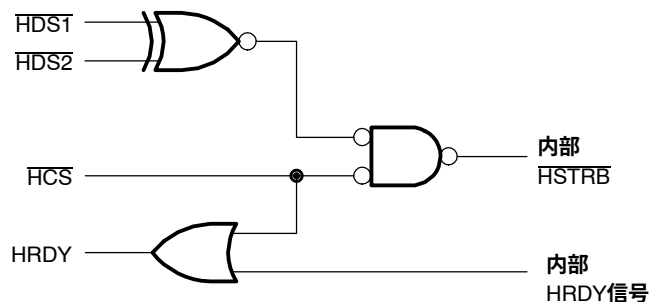
HPI-16の主な特長は、次のとおりです。

- ☐ ホスト・アドレス・バスに直接接続できる16ビット・アドレス・バス('VC5420上の18ビット・アドレス・バス)
- ☐ バイトの順序づけを必要としない16ビット・データ・バス
- ☐ 各種の16ビット・ホストとの接続に適したストローブ信号と制御信号を含む、柔軟なインターフェイス
- ☐ 共用モードと分離モードの選択により、さらに柔軟なインターフェイスが可能
- ☐ ダイレクト・メモリー・アクセス(DMA)コントローラと同期したメモリー・アクセスにより、内部メモリーの全アドレス領域に対するアクセスが可能
- ☐ HRDYピンのソフトウェアによるポーリング
- ☐ データ・フェッチのソフトウェア制御
- ☐ ストローブ信号および制御信号

インターフェイスをとるために2つのストローブ入力とHPIセレクト・ピンがあります。これら3つの入力ピンは、HPIのストローブを制御します。内部ストローブは、HPIへの実際のストローブ信号(HSTRB)として機能します。

図5-1に示すように、ストローブの状態は3つのストローブ入力、 $\overline{\text{HCS}}$ 、 $\overline{\text{HDS1}}$ 、および $\overline{\text{HDS2}}$ の状態によって決定されます。 $\overline{\text{HCS}}$ (HPIチップ・セレクト・ピン)は、 $\overline{\text{HDS}}$ ピンのストローブがアクティブの時はローである必要があります。 $\overline{\text{HCS}}$ がハイのままだと、 $\overline{\text{HDS}}$ ピンの状態は無視され、 $\text{HRDY}$ がハイになります。 $\text{HRDY}$ 信号は、現在の転送が完了したかどうかに関係なくホストにフィードバックします( $\text{HRDY}$ がハイの状態)。したがって、ホストのアクセス速度がHPIのアクセス速度よりはるかに遅く、 $\text{HRDY}$ が必要ない場合は、 $\overline{\text{HCS}}$ および片方の $\overline{\text{HDS}}$ ストローブ入力を直結できます。これによって事実上HPIのチップ・セレクトとストローブが同時に行われます。

図5-1. HPIストローブおよび選択ロジック



5

ホストと接続する方法はいくつかありますが、すべての接続方式は2つの動作モード、共用モードと分離モードのどちらかに分類されます。共用モード動作では、ホスト・プロセッサが $\text{HCNTL0}$ と $\text{HCNTL1}$ という2つの制御信号によってHPIの動作を制御し、それに対して、分離モードでは、HPIアドレス・バスとデータ・バスを使ってデータのリード/ライトが行われます。このHPIのモードは、 $\text{HMODE}$ ピンを使ってハードウェア的に設定できます。 $\text{HMODE} = 0$ の場合は共用モードが選択されます。

$\text{HMODE}$ をサポートするデバイスと、サポートしないデバイスがありえます。各デバイスのデータ・シートを参照して、これらのモードがサポートされているかどうか確認してください。表5-1に、全ての使用可能なHPI-16ピンと、それぞれの機能についての説明があります。

## HPI-16の動作の概要

表5-1. HPI-16ピンの説明

| HPI-16<br>信号                           | I/O/Z状態 <sup>†</sup> | ホストへの接続                            | 信号の説明                                                                                                                                                                                                                                                                                                                         |
|----------------------------------------|----------------------|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| HMODE                                  | I                    | 通常は一定値であり、ハイまたはローに固定されます。          | HPIモード・セレクト・ピン。HMODEのロジック・レベルによってHPIの動作モードが決まります。このピンにローが入力されている場合は共用モードが選択されます (HCNTL0/1が必要)。一方、ハイが入力されている場合は分離モードが選択されます。このピンは一部のデバイスでのみ使用できます。HMODEがサポートされているかどうか調べるには、各デバイスのデータ・シートを参照してください。                                                                                                                             |
| HAS                                    | I                    | ALE – アドレス・ラッチ・イネーブルまたはアドレス・ストローブ。 | 共用モードのみで使用されます。アドレス/データ・バスを共用しているホストは、このピンをALEに接続します。この入力信号の立ち下がりエッジにより、HR/ $\overline{W}$ ピン、HCNTL0ピン、およびHCNTL1ピンのロジック・レベルをラッチします。HAS信号が使用される場合は、HASはストローブ (HCSまたはHDS) がアクティブになる前にアクティブになる必要があります。アドレス/データ・バスが分離しているホストは、この信号をハイに固定する必要があります。その結果、HR/ $\overline{W}$ 入力、HCNTL0入力、およびHCNTL1入力はHCS = 0のときHDSの立ち下がりエッジでラッチされます。 |
| $\overline{HCS}$                       | I                    | アドレス・ラインまたはセレクト・ライン                | HPIチップ・セレクト。HPIセレクト入力の役割を果たします。HPIモジュールを選択するには、HCSがローである必要があります。HCSは、アクセス中もローのままではなりません。HCSは通常、HDSストローブがアクティブになる前にアクティブになりますが、HDSに接続して、セレクトとストローブを同時に行うこともできます。内部HPIストローブ・ロジックの詳細については、図5-1を参照してください。                                                                                                                         |
| $\overline{HDS1}$<br>$\overline{HDS2}$ | I                    | リード・ストローブ、ライト・ストローブまたはデータ・ストローブ    | HPIデータ・ストローブ・ピン。HPIモジュールのデータ入出力を、ストローブするために使用されます。データの方法は、HR/ $\overline{W}$ 信号のロジック・レベルによって決まります。HDS信号は、その立下りエッジで制御信号をラッチすることにも使われます (HASがハイに固定されている場合)。HPIDライト・アクセス中は、データがHDSの立ち上がりエッジでHPIDレジスタ内にラッチされます。リード・アクセス中は、これらのピンがデータ・バスへの出力をイネーブルにします。                                                                            |
| HR/ $\overline{W}$                     | I                    | R/ $\overline{W}$ ストローブ            | HPIリード/ライト信号。 $\overline{HAS}$ または $\overline{HDS}$ の立ち下がりエッジで、現在のアクセスがリードかライトかをHPIに通知します。ハイの場合は転送がHPIからのリード動作であることを示し、ローの場合はHPIへのライト動作を示します。                                                                                                                                                                                 |

<sup>†</sup> I = 入力、O = 出力、Z = ハイ・インピーダンス



表5-1. HPI-16ピンの説明 (続き)

| HPI-16           |                      |                  |                                                                                                                                                                                                                                                               |
|------------------|----------------------|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 信号               | I/O/Z状態 <sup>†</sup> | ホストへの接続          | 信号の説明                                                                                                                                                                                                                                                         |
| HCNTL0<br>HCNTL1 | I                    | アドレス・ラインまたは制御ライン | HPIアクセス制御入力。これらのロジック・レベルは、HASまたはHDSの立ち下がりエッジでラッチされます。これら4つのピンの状態により、現在の転送のアクセス・モード (HPIC、HPID(インクリメント有)、HPIA、およびHPID(インクリメント無))が決定されます。                                                                                                                       |
| HA[n:0]          | I                    | ホストのアドレス・バス      | HPIアドレス・バス。HPIを分離モードで動作させる場合は、(n+1)本のピンをホストのアドレス・バスに接続する必要があります。しかし、ホストがアドレス/データ・バスを共用していて、共用モードでHCNTL0/1ピンを使用する場合は、これらのピンを一定のロジック・レベルに固定することができます。nの値は、DSPのアドレス・レンジによって変動します。たとえば、VC5420には18ビットのアドレス・レンジ(A17:A0)があるため、n = 17となります。                           |
| HD[15:0]         | I/O/Z                | ホストの16ビット・データ・バス | HPIデータ・バス。HPIデータ・バスは16ビット幅で、HPIに対しデータを入出力します。これらのピンは、EMU1/OFFがアクティブ(ロー)のとき、またはリード・アクセスが行われていないときはハイ・インピーダンスになります。                                                                                                                                             |
| HRDY             | O/Z                  | 非同期レディー入力        | HPIレディー信号。ハイの場合は、現在の転送が完了したことを示します。ローの場合は、HPIが次のアクセスに対してレディー状態でないことを示します。ホストはHRDYがハイになるまで待つ必要があります。このロジック・レベルは、HPICレジスタのHRDYビットをリードすることで、ソフトウェア・ポーリングすることができます。HCSがハイになると、HRDY信号は強制的にハイになります。EMU1/OFFがアクティブ(ロー)のときはこのピンはハイ・インピーダンスになります。                      |
| HINT             | O/Z                  | ホスト割り込みピン        | ホスト割り込み。DSPIはHPICレジスタのHINTビットに1をライトすることで、ホスト・プロセッサに対し割り込みを発生できます。後続のHINT割り込みが発生する前に、ホストがHPICレジスタのHINTビットに1をライトして前の割り込みをクリアする必要があります。DSPがリセット状態のときは、このピンがハイにドライブされます。このピンはアクティブ・ローであり、HPICレジスタのHINTビットの値が反転されて現れます。EMU1/OFFがアクティブ(ロー)のときはこのピンがハイ・インピーダンスになります。 |
| HPIRS            | I                    | 制御ピン、またはハイ/ローに固定 | HPIリセット・ピン。HPIをリセットします。HPIアクセスは行われず、HD[15:0]がハイ・インピーダンス状態になります。<br><br>このピンは一部のデバイスでのみ使用できます。HPIRSがサポートされているかどうか調べるには、各デバイスのデータ・シートを参照してください。                                                                                                                 |

<sup>†</sup> I = 入力、O = 出力、Z = ハイ・インピーダンス

表5-1. HPI-16ピンの説明 (続き)

| HPI-16 |                      |                         |                                                                                                                                                                                              |
|--------|----------------------|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 信号     | I/O/Z状態 <sup>†</sup> | ホストへの接続                 | 信号の説明                                                                                                                                                                                        |
| HPIENA | I                    | 通常はVccに接続               | <p>HPIイネーブル。HPIENAピンは、HPIを完全に止めるために使用されます。このピンをグラウンドに接続すると、HPIが停止し、電力が消費されません。HPIを使用する場合は、このピンを+Vccに接続します。</p> <p>このピンは一部のデバイスのみ使用できます。HPIENAがサポートされているかどうか調べるには、各デバイスのデータ・シートを参照してください。</p> |
| HPI16  | I                    | 通常は一定値であり、ハイまたはローになります。 | <p>HPI16セレクト・ピン。このピンにより、HPIが8ビット・ホスト(0)または16ビット・ホスト(1)をサポートできます。8ビット動作の詳細については、第4章を参照してください。</p> <p>このピンは一部のデバイスでのみ使用できます。HPI16がサポートされているかどうか調べるには、各デバイスのデータ・シートを参照してください。</p>               |

<sup>†</sup> I = 入力、O = 出力、Z = ハイ・インピーダンス

## 5.2 共用モード

共用モードでは、HPIがHCNTL0/1ピンによって制御され、ホスト/DSP割り込みを必要とするホストか、アドレス・ラインおよびデータ・ラインを共用しているホストに対し使用されます。ホストは、2つの方法でHCNTL0/1ピンとHR/ $\overline{W}$ ピンを制御できます。ひとつは、ホストが専用出力またはアドレス・ピンを使ってHCNTL0/1ピンとHR/ $\overline{W}$ ピンをドライブする場合です。この場合、HCNTL0/1とHR/ $\overline{W}$ のロジック・レベルはホストによりドライブされるストローブ( $\overline{HDS}$ )信号の立ち下がりエッジでラッチされます。もう一つは、ホストがアドレス・バスおよびデータ・バスを共用している場合です。専用アドレス・バスがないため、ホストはデータ・アクセス用のバスを使って制御信号を駆動します。HCNTL0/1信号とHR/ $\overline{W}$ 信号はHPIの $\overline{HAS}$ 信号を使ってラッチされ、これにより、ホストはその後にライト動作またはリード動作を実行する時間を得ることができます。図5-2は、共用モードで $\overline{HAS}$ を使ってHPIを制御するホストの一般的なブロック図を示しています。

5

図5-2. 共用モードでのHPI-16へのインターフェイス('VC5420)

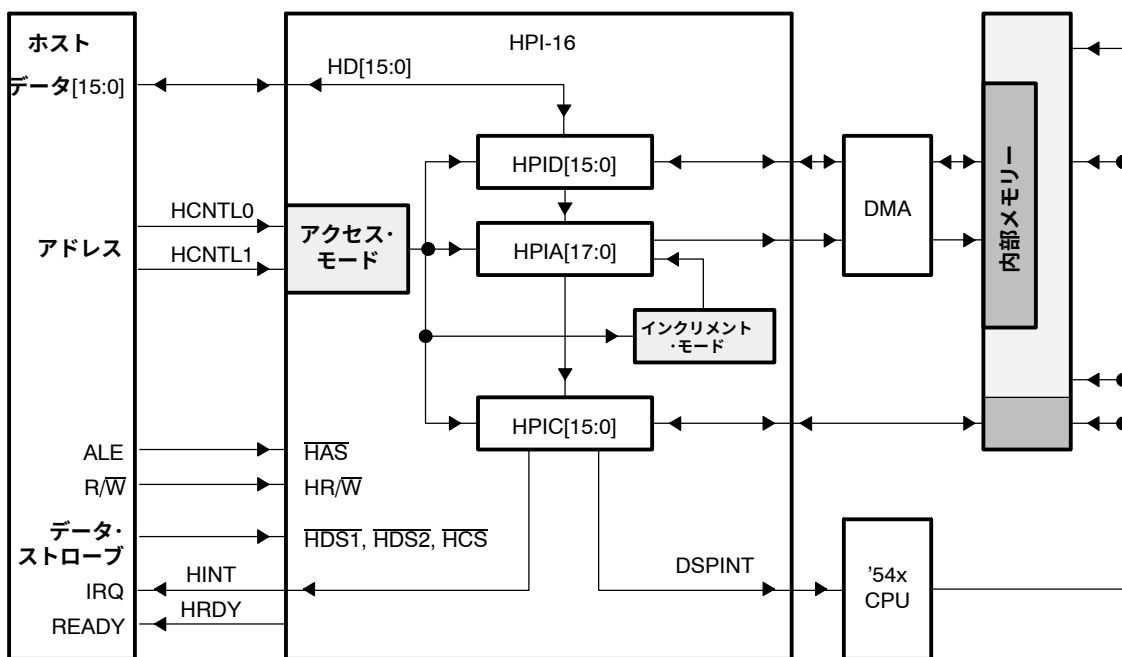


図5-2では、HCNTL信号とHR/ $\overline{W}$ 信号はアドレス・バスおよびR/ $\overline{W}$ によってドライブされ、 $\overline{HAS}$ に接続されたアドレス・ラッチ・イネーブルによってラッチされます。図5-2と表5-2は、HCNTL0/1信号がどのようにHPIを制御するかを示しています。HPIAレジスタを一度初期化した後、連続するデータ・メモリーへのアクセスに対してHPIAインクリメント・モードを使うと高速なデータ転送が実現します。

## 共用モード

表5-2. HCNTL0/1モード

| HCNTL1 | HCNTL0 | ホスト・アクセス・モード                                                        |
|--------|--------|---------------------------------------------------------------------|
| 0      | 0      | ホストはHPICレジスタのリード/ライトができます。                                          |
| 0      | 1      | ホストはHPIDレジスタのリード/ライトができます。HPIAは、HPIアクセスが実行されるたびに自動的にポストインクリメントされます。 |
| 1      | 0      | ホストはHPIAレジスタへのリード/ライトができます。                                         |
| 1      | 1      | ホストはHPIDレジスタへのリード/ライトができます。HPIAレジスタは変更されません。                        |

## 5

ホスト・ポート・インターフェイス制御レジスタ(HPIC)は、HPIおよびDSPがアクセスできる内部設定レジスタです。このレジスタはHPIおよびDSPに対し、DSPからホストへの割り込みおよびホストからDSPへの割り込みと、HRDYピンのソフトウェア・ポーリングの実行を提供します。DSPはこのレジスタ(データ領域のアドレス0x2cに割り当てられています)に常にアクセスできますが、ホストはHCNTLモードが(00)の場合にしか、このレジスタにアクセスできません。

図5-3. HPICレジスタ

|      |       |       |      |      |        |      |
|------|-------|-------|------|------|--------|------|
| 15-6 | 5     | 4     | 3    | 2    | 1      | 0    |
| RSVD | XHPIA | FETCH | HRDY | HINT | DSPINT | RSVD |

HPICレジスタには、ホスト・プロセッサおよびDSPがアクセスできます。表5-3は、各ビット/フィールドとその説明を示しています。

表5-3. HPICビットの説明

| ビット    | アクセス元 |       | 説明                                                                                                                                                                        |
|--------|-------|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|        | ホスト   | DSP   |                                                                                                                                                                           |
| DSPINT | R-0/W | R-0   | ホストからDSPへの割り込み。ホストはDSPINTに1をライトすることでDSPに対して割り込みを発生することができます。このビットは、ホストおよびDSPによって常に0がリードされます。DSPがこのビットにライトしても影響はありません。                                                     |
| HINT   | R/W-1 | R/W-1 | DSPからホストへの割り込み。DSPはHINTビットに1をライトしてホストに対する割り込みを発生します。<br><br>HINTビットの値を、HINTピンでは反転してドライブします。ホストは、HINTピンの状態をクリアするためにHINTに1をライトする必要があります。ホストまたはDSPがHINTビットに0をライトしても影響はありません。 |
| HRDY   | R     | R     | HRDYピンのロジック・レベルがこのフィールドに反映されます。ホストとDSPはHRDYピンのソフトウェア・ポーリングのためこのビットをリードすることができます。HRDY = 0の場合、HPI-16は現在のデータ・アクセスを完了していません。                                                  |

表5-3. HPICビットの説明 (続き)

| ビット   | アクセス元 |     | 説明                                                                                                                                                                                                               |
|-------|-------|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | ホスト   | DSP |                                                                                                                                                                                                                  |
| FETCH | R-0/W | R-0 | ホスト・データ・フェッチ要求。ホストがこのビットに1をライトすると、現在のHPIAアドレスにあるデータがフェッチされ、HPIDレジスタにロードされます。このビットはホストおよびDSPによって常に0がリードされます。                                                                                                      |
| XHPIA | R/W   | R   | 拡張アドレス・イネーブル。XHPIA = 1の場合、HPIAレジスタへのホストによるライトは上位ビットHPIA[n:16]にロードされます。XHPIA = 0の場合、HPIAへのホストによるライトはHPIA[15:0]にロードされます。全てのn+1個のアドレス・ビットは、自動インクリメント・モードの場合、インクリメントされます。HPIAレジスタのリードも同様に行われます。ホストのみがこのビットにアクセスできます。 |

HPIAレジスタは、次のデータ・アクセスを行うアドレスを含みます。HCNTOL0/1 = 10bのモードでは、HPIAレジスタが自動的にインクリメントされます。FETCH = 1の場合は、HPIAレジスタが指すデータのリード要求を行います。FETCHビットをHRDYビットとともに使用してソフトウェア・ポーリングを実行し、内部アクセスがいつ完了するかを判断することができます。FETCHビットとHRDYビットは、HPIDレジスタを更新したい場合に有効です。FETCHビットは常に0としてリードされます。FETCHを実行しても、HPIAレジスタはインクリメントされません。

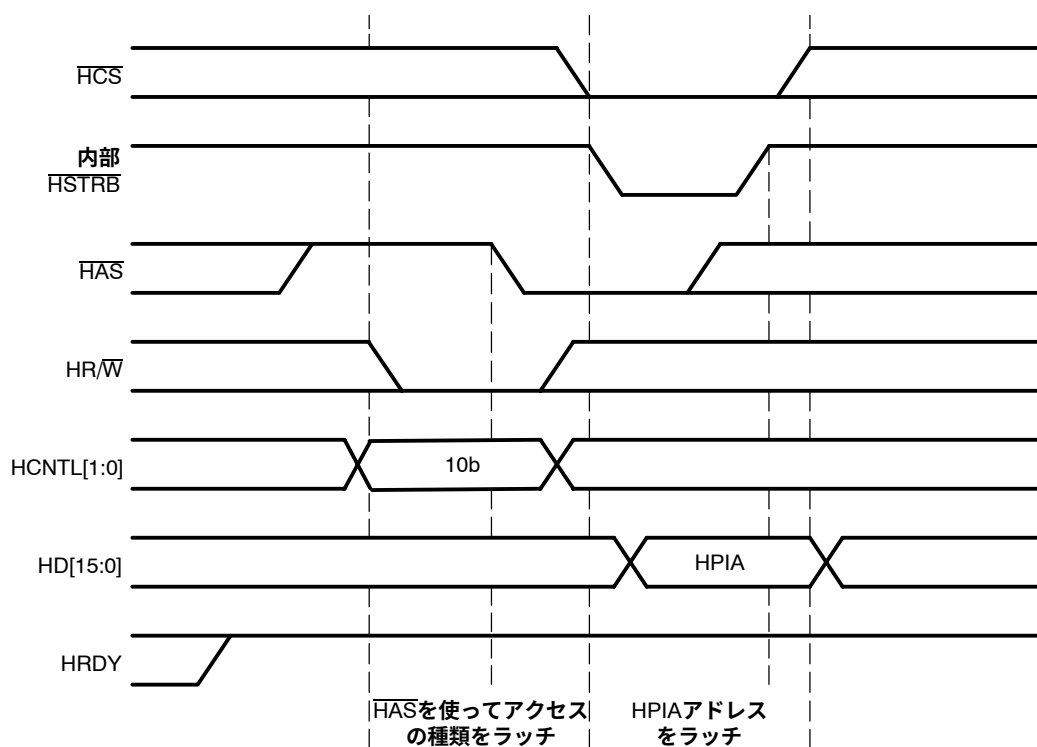
ホストとDSPの間の割り込みは、HPICレジスタのDSPINTビットとHINTビットをセットし、ポーリングすることで可能になります。DSPINTビットは、ホストによって1にセットされると、DSPへの割り込みを通知します。DSPはIMRレジスタで割り込みをイネーブルにすることで、その割り込み処理を実行できます。DSPはHINTビットに1をライトすることでホストへの割り込みを発生します。この場合、ホストはHINTビットへ1のライトを実行して前のHINT要求を認識する必要があります。DSPINTビットは常に0としてリードされます。HINTビットとDSPINTビットに0をライトしても影響はありません。

XHPIAビットの値により、HPIAのどのアドレス・ビットが初期化されるかが決まります。データ・バスは16ビットであり、アドレス・レンジは16ビットより大きい(VC5420では18ビット)ため、XHPIAビットがゼロにセットされるとHPIAの下位16ビットがロードされ、XHPIA = 1の場合は拡張アドレス・ビット(>16)がロードされます。データ・アクセスを実行する前に、すべてのアドレス・ビットを初期化して下さい。HPIAレジスタの内容をリードするには、同じようにXHPIAを変更する必要があります。

### 5.2.1 $\overline{\text{HAS}}$ を使用するホスト・アクセス

ホスト・アドレス・ストローブ( $\overline{\text{HAS}}$ )は共用モードでのみ有効です。 $\overline{\text{HAS}}$ 信号はアドレス・バスおよびデータ・バスを共用するホスト・プロセッサに対して追加ロジックを必要としないインターフェイスを可能にします。HCNTL0/1およびHR/ $\overline{\text{W}}$ の状態をHPIにラッチするには、 $\overline{\text{HAS}}$ の立ち下がりエッジが使用されます。まず、ホストがHCNTL0/1ピンおよびHR/ $\overline{\text{W}}$ ピンを希望のアクセス・モードにドライブして、HPIの転送モードがラッチされます。 $\overline{\text{HAS}}$ はHCS入力の影響を受けません。したがって、ホストは後続のアクセスを実行するための時間を確保できます。HDSがローにドライブされた後、 $\overline{\text{HAS}}$ 信号がハイになることがあります。これは、すぐにデータ・アクセスが行われることを示します。 $\overline{\text{HAS}}$ はハイである必要はありませんが、アクセスの種類に変更があるときは最終的にハイにする必要があります。図5-4に、 $\overline{\text{HAS}}$ を使用したライト・アクセスを示します。

図5-4.  $\overline{\text{HAS}}$ を使用するHPIAライト

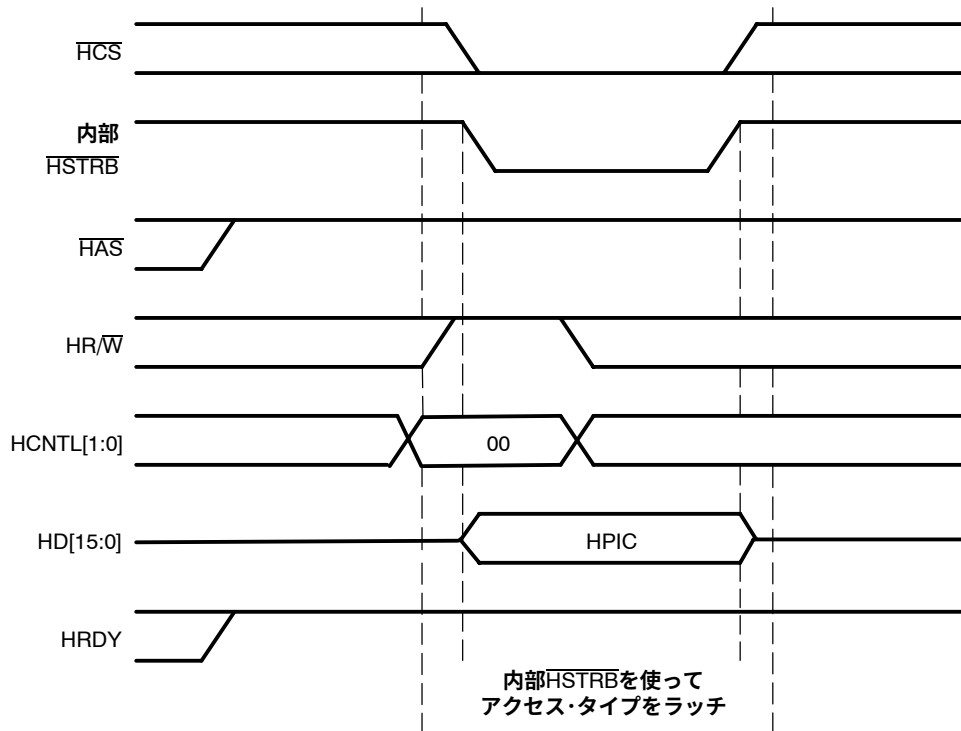


### 5.2.2 $\overline{\text{HAS}}$ を使用しないホスト・アクセス

HCNTL0/1ピンをドライブできる専用信号(またはビットI/O)をホスト・プロセッサが備えている場合は、 $\overline{\text{HAS}}$ 信号は必要ありません。それらの専用信号はHCNTL0/1ピンおよびHR/ $\overline{\text{W}}$ ピンに直接接続できます。これによって、 $\overline{\text{HAS}}$ によるHCNTL0/1データのセットアップおよびストロブが必要のないアクセスを容易に行うことができます。たとえば、 $\overline{\text{HAS}}$ を使用しないHPICリードは、HCNTL0/1の値(00)をドライブした後、HPIをストロブすれば実行されます。内部 $\overline{\text{HSTRB}}$ 信号の立ち下がりエッジの後、HPIがHPICレジスタのデータをデータ・バス上にドライブします。図5-5は、 $\overline{\text{HAS}}$ 信号を使用しないリード・アクセスのタイミングを示しています。

図5-5.  $\overline{\text{HAS}}$ を使用しないHPICリード

5



### 5.2.3 自動インクリメント動作

すべてのHPIは、データ転送速度を上げるために自動インクリメント機能を備えています。自動インクリメント機能は、現在のアクセスが完了した後、データをプリフェッチし、HPIA内のアドレスを次の値に変更します。この機能によりHPIAレジスタが自動的に変更されるため、ホストはHPIAレジスタの内容を変更する必要なく、連続するメモリー空間に対して連続的なHPIリード/ライト転送を実行できます。この機能は、HCNTL0/1モード = 10bのときイネーブルになります。

図5-6. 自動インクリメントを使用するHPIDリード

5

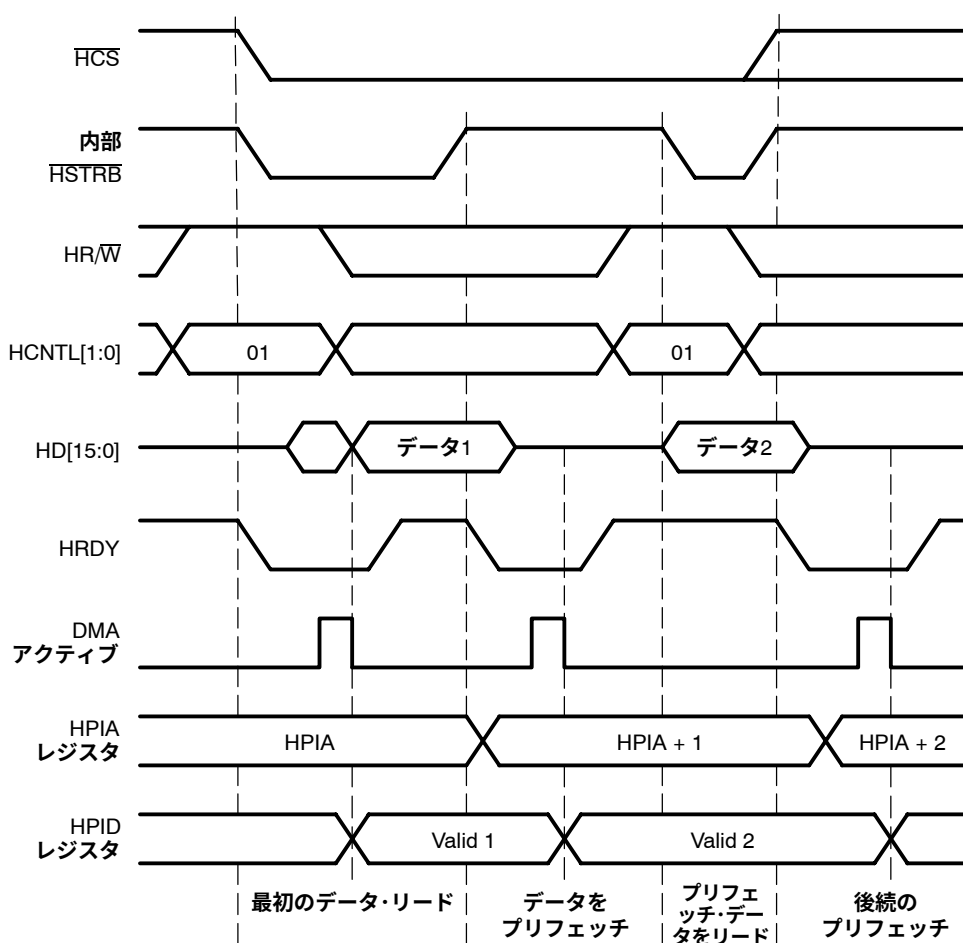
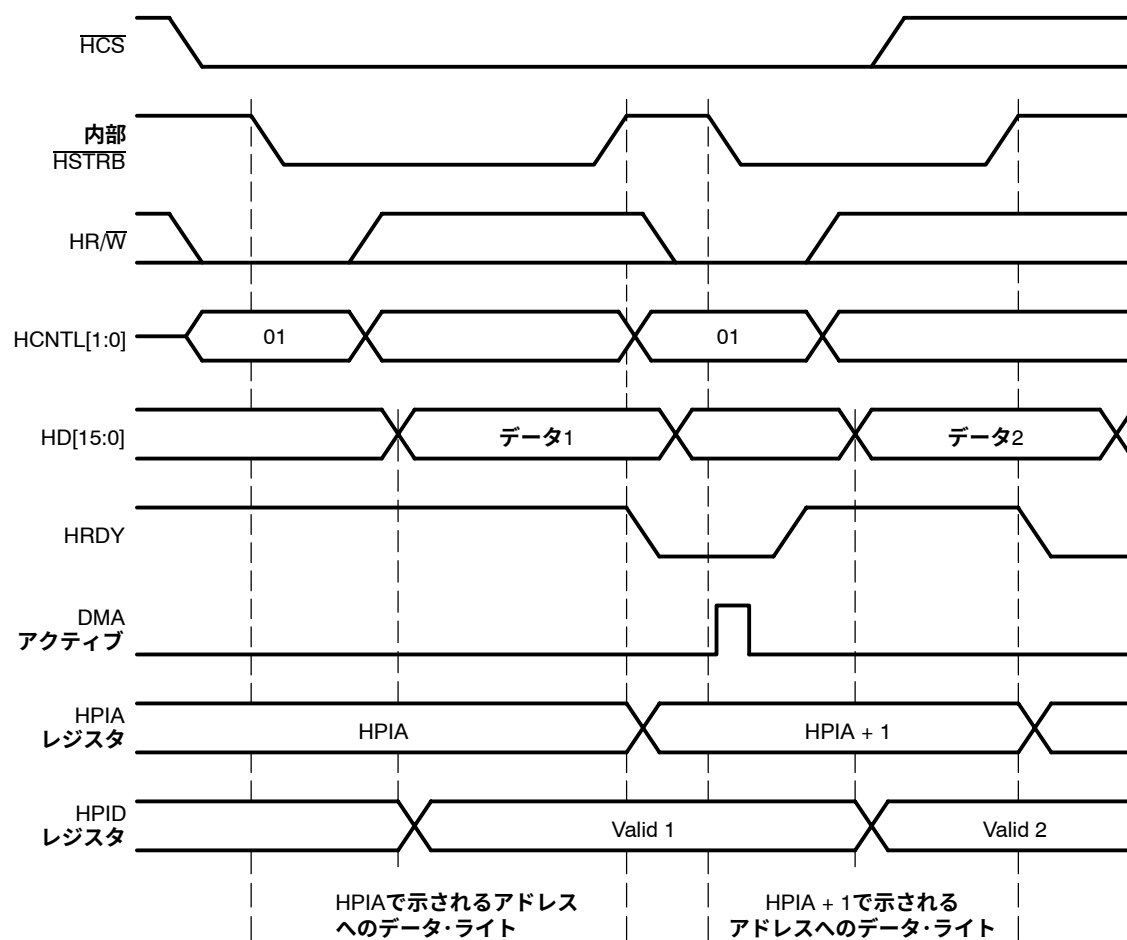




図5-7. 自動インクリメントを使用するHPIDライト



自動インクリメントありのリードを実行した場合、プリフェッチ特性により、無効な(古い)データがホストCPUによってリードされることがあります。これは、ホストがリード・アクセスを実行した後、DSPが次のアドレスにあるデータを更新した場合に起こる可能性があります。自動インクリメント機能およびプリフェッチの構造上、ホストによって次にリードされるデータは最新データ値ではない可能性があります。DSPがライトしている領域をホストがリードする場合は、データをリードする前にFETCHを実行してください。FETCHは、HPICレジスタのFETCHビットをセットして実行します。FETCH実行後は、HPIからデータをリードする前にHPICレジスタをリードし、HRDYビットをポーリングします。図5-6と図5-7は、それぞれ自動インクリメント機能を使用したリード動作とライト動作を示しています。最初のリード(自動インクリメント使用)と、自動インクリメント機能を使用しないすべてのリードでは、アクセスが開始されると(内部 $\overline{\text{HSTRB}}$ の立ち下がりエッジ)HRDY信号がインアクティブになります。インクリメント・モードでの連続するリードにより、データのプリフェッチが実行されます。これによってHPIの転送速度が向上します。

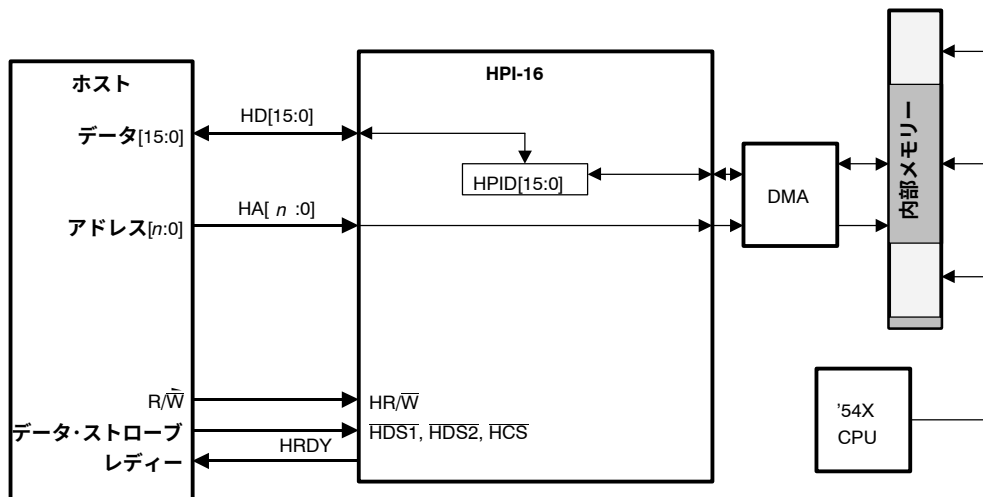
ライト動作は常に内部 $\overline{\text{HSTRB}}$ 信号の立ち上がりエッジで実行されます。連続ライト時に、HRDYのnot-ready状態を使って次のライト動作を開始すると、ライト動作の処理速度を改善できます。ホストは次のライトを実行する前に $\overline{\text{HSTRB}}$ をハイに遷移させ、少なくとも10nsの間ハイに保つ必要がありますが、HRDYがnot-readyである間にはこのホールド時間は満たされます。これは図5-7に示されています。

### 5.3 分離モード

分離モードでは、アドレス・バスとデータ・バスが分離しているホストが、16ビット双方向データ・バスを使用してHPIデータ(HPID)レジスタに、nビットのアドレス・バスを使用してHPIAアドレス・レジスタ(HPIA)にアクセスできます。分離モードではHPICレジスタは使用されません。したがって、ホストとDSPの間で割り込み(DSPINTおよびHINT)を発生することはできません。ホスト・アドレス・バスはHPIAのドライブに使用できるため、自動インクリメント機能は必要なく、FETCHビットとXHPIAビットも必要ありません。HRDYのソフトウェア・ポーリングをすることはできませんが、HRDYピンは完全に機能し、DSPへのアクセスが内部で完了するまで後続のホスト・アクセスを保留することができます。HRDYピンは、共用モードの節で説明したとおりに機能します。図5-8は、分離モードでのHPIへのインターフェイス方法を示しています。

5

図5-8. 分離モードでのHPI-16へのインターフェイス('VC5420)



HPIセレクト・ピン( $\overline{HCS}$ )およびストローブ信号( $\overline{HDS1}$ および $\overline{HDS2}$ )は、共用モードの節での説明と同様に機能します。 $\overline{HAS}$ ピン、HCNTL0ピン、およびHCNTL1ピンは分離モードでは使用されません。

図5-9および図5-10に、分離モードでの一般的なアクセスを示します。データ転送速度は、共用モードで $\overline{HAS}$ 信号や自動インクリメント機能を使用しない場合と同じです。DMAコントローラと同時に使用した場合のバスの割り当ては、共用モードと同様に処理されます。

## 分離モード

図5-9. 分離モードでのHPIDリード

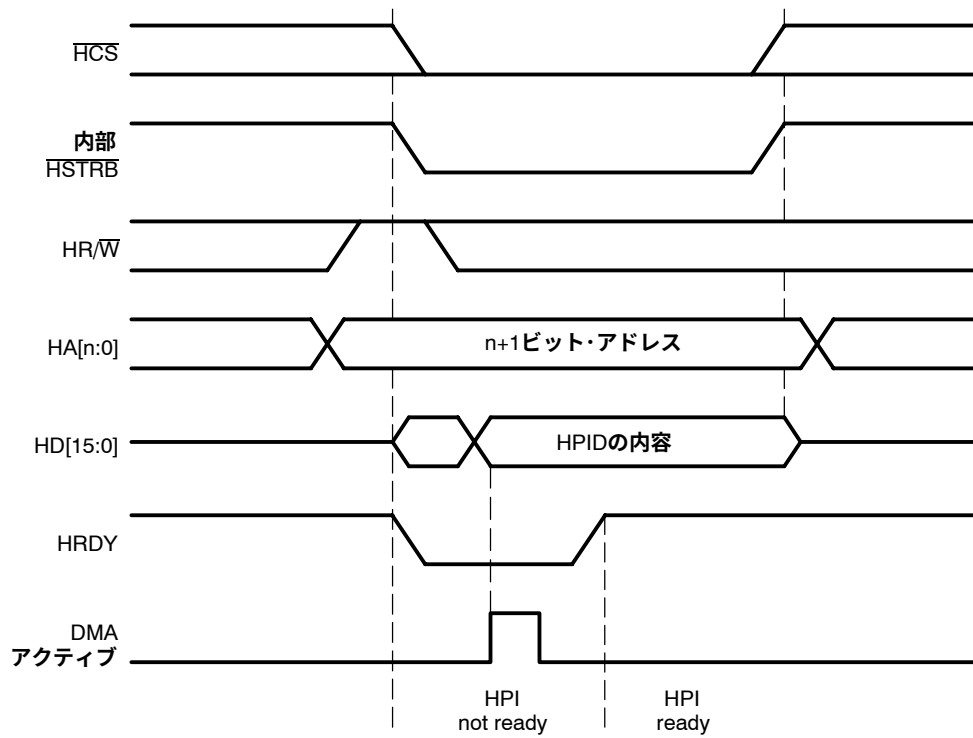
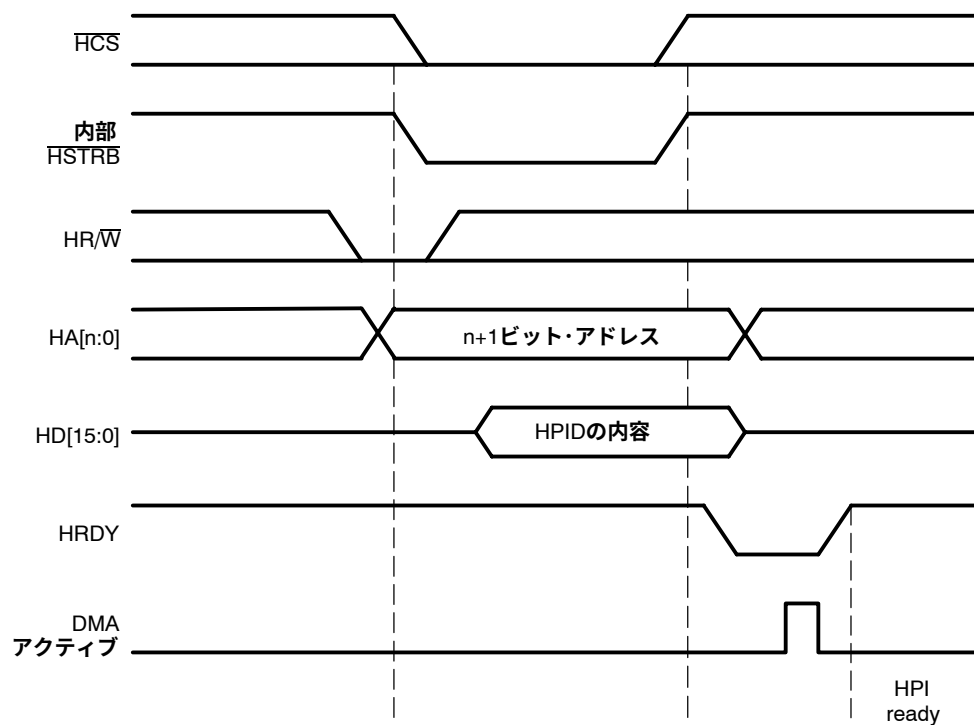


図5-10. 分離モードでのHPIDライト



## 5.4 HPI-16メモリー・マップ

HMODEピンのロジック・レベルにより、HPI-16はDSPメモリー('VC5420では内部メモリーのみ)へのアクセスを共用モードまたは分離モードで実行できます。前述したように、共用モードではメモリー・マップへのアクセスにHPIAレジスタが使用されます。分離モードではnビットのHPIアドレス・バスを使用して直接メモリー・マップへアクセスします。どちらのモードでも、DMAバス経由でオンチップ・メモリーの全領域にアクセスできます。図5-11は、'VC5420のHPIメモリー・マップを示しています。この'VC5420は、18ビットのアドレス・バスを持つデバイスです。

図5-11. HPIに対する'VC5420のメモリー・マップ

| プログラム・ページ0 |                                  | プログラム・ページ1 |                                  | プログラム・ページ2 |                                  | プログラム・ページ3 |                                  |
|------------|----------------------------------|------------|----------------------------------|------------|----------------------------------|------------|----------------------------------|
| Hex        |                                  | Hex        |                                  | Hex        |                                  | Hex        |                                  |
| 0000       | 予約                               | 10000      | 予約                               | 20000      | 予約                               | 30000      | 予約                               |
| 001F       |                                  |            |                                  |            |                                  |            |                                  |
| 0020       | McBSP DXR/DRR MMRのみ              | 005F       |                                  | 005F       |                                  | 005F       |                                  |
| 0060       | オンチップ DARAM 0 (オーバーレイ) プログラム/データ | 0060       | オンチップ DARAM 0 (オーバーレイ) プログラム/データ | 0060       | オンチップ DARAM 0 (オーバーレイ) プログラム/データ | 0060       | オンチップ DARAM 0 (オーバーレイ) プログラム/データ |
| 3FFF       |                                  | 13FFF      |                                  | 23FFF      |                                  | 33FFF      |                                  |
| 4000       | オンチップ SARAM 1 (オーバーレイ) プログラム/データ | 14000      | オンチップ SARAM 1 (オーバーレイ) プログラム/データ | 24000      | オンチップ SARAM 1 (オーバーレイ) プログラム/データ | 34000      | オンチップ SARAM 1 (オーバーレイ) プログラム/データ |
| 7FFF       |                                  | 17FFF      |                                  | 27FFF      |                                  | 37FFF      |                                  |
| 8000       | オンチップ SARAM 2 (32Kワード) プログラム/データ | 18000      | オンチップ SARAM 3 (2Kワード) プログラム      | 28000      | 予約                               | 38000      | 予約                               |
|            |                                  |            |                                  | 2EFFF      |                                  |            |                                  |
|            |                                  |            |                                  | 2F000      | オンチップ SARAM 4 (32Kワード) プログラム     |            |                                  |
| FFFF       |                                  | 1FFFF      |                                  | 2FFFF      |                                  | 3FFFF      |                                  |

注: I/Oスペースはアクセスできません

すべての内蔵メモリーは、P2の4Kワード・ブロック(0x2F000-0x2FFFF)を除き、8Kブロックに分割されます。DROMビットとOVLYビットはメモリー・マップに影響を与えません。

HPIは、CPUによって制御されるメモリー設定ビットの影響を受けません。そのため、OVLYビットとDROMビットの値は、HPIに対するメモリー・マップに影響を与えません。'VC5420の場合、内部DSPメモリーへのアクセスは常に、プログラムおよびデータがオーバーレイされているものとみなされます。プログラム空間とデータ空間を区別する方法はありません。I/Oスペースへは、HPIによるアクセスはできません。

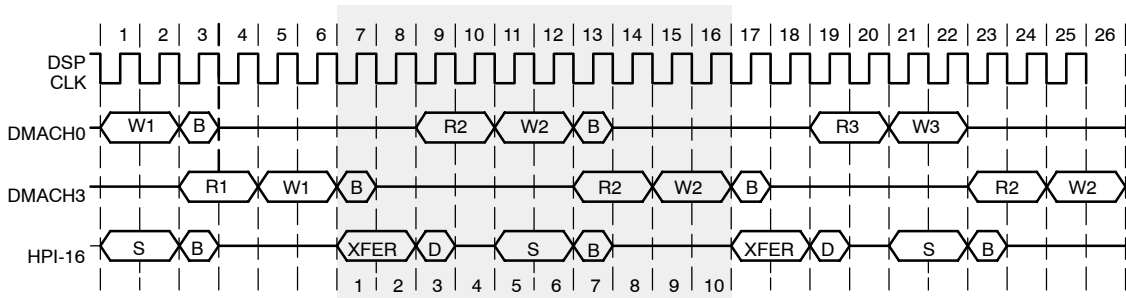
その他のデバイスに関するHPIメモリー・マップおよび設定については、各デバイスのデータ・シートを参照してください。

### 5.5 HPI-16とDMAの相互作用

HPI-16は、全ての内部メモリーへのホスト・アクセスを可能にするDMAコントローラと共に動作します。ホストCPUがアクセスを要求するたび、DMAはその時点での転送完了後に、HPIに対してDMAバスを解放します。そのため、他のDMA転送が行われている場合、HPIアクセスにレイテンシーが発生します。ホストは、他のDMAチャンネルが動作中でなければ、最大6-DSPサイクルごとに1回の割合でDSPのメモリーにアクセス(ライト)できます。しかし、2つ以上のDMAチャンネルが動作している場合は、HPIの転送速度は10サイクルに低下します。HPIは、DSPのCPUを含めて最も優先度が高くなっています。その結果、CPUまたは他のDMAチャンネルが同じメモリー位置で競合しても、HPIにアクセスが許可されます。図5-12は、DMAコントローラが同時に動作する場合のHPIのアクセス・タイミングが示されています。

5

図5-12. HPIとDMAの相互作用



注： S = DSPクロックへの同期  
 B = DMAバスに対する要求  
 D = 遅延  
 Rn = n回目のアクセス時のリード・サイクル  
 Wn = n回目のアクセス時のライト・サイクル

1つまたは複数のDMAチャンネルが動作している場合、HPI転送のレイテンシーは長くなります。しかし、HPIのレイテンシーが、動作中の一つのDMAチャンネルによるレイテンシーよりも長くなることはありません。HPIバスの許可は、DMAチャンネルの転送が切り替わる間に割り当てられます。そのため、HPIによるデータ転送が行われる場合は、他のDMAチャンネルでのデータ転送速度が犠牲になります。HPIが介入しない場合も、高優先順位のDMAチャンネルが連続して動作した場合、低優先順位のDMAチャンネルは待機します。各種C54xデバイスのHPIリードおよびライト転送のレイテンシーについては、各デバイスのデータ・シートを参照してください。

HPIAレジスタは、DMAコントローラによるレイテンシーの影響を受けません。HPIAレジスタはHPIペリフェラルの一部であるため、HPIAレジスタへのアクセスにはDMAバスが使用されません。DSPINTビットおよびHINTビットをゼロにセットする場合は、HPICレジスタへのアクセスにはレイテンシーが発生しません。これらのアクセスでは、後続のアクセスを即座に開始できるため、HRDYがローにドライブされることはありません。しかし、DSPINTビットまたはHINTビットに1をセットすると、DSPのCPUへのアクセスが必要になるため、レイテンシーが発生します。その結果として、HRDY信号は3-DSPサイクルの間ローになります。



## 5.6 リセット時のHPI-16の動作('VC5420のみ)

'VC5420のHPI-16は、DSPのCPUがリセット状態のときも動作できます。'VC5420では、RESET信号がコアCPUとHPIの間で分離されます。専用HPIリセット・ピン $\overline{\text{HPIRST}}$ がゼロの場合は、HPIがリセット状態になります。 $\overline{\text{HPIRST}}$ 信号が1のときは、HPIはDSPの全ての内部メモリー空間にアクセスできます。この場合でも1回のアクセスには6-DSPクロック・サイクルが必要であるため、最大データ転送速度は100MIPS動作時で、33MB/sとなります。

ホスト・プロセッサは、 $\overline{\text{RS}}$ ピンを制御することでDSPをリセットできます。この場合、ホストはその時点でのメモリー・アクセスを完了し、6-DSPクロック・サイクル待つから $\overline{\text{RS}}$ ピンをローにドライブする必要があります。リセット時でも、ホストは内部メモリーにアクセスできます。DSPのリセットを解除するために、ホストが $\overline{\text{RS}}$ ピンをハイにドライブできますが、後続のメモリー・アクセスを実行する前に少なくとも20-DSPクロック・サイクル待つ必要があります。

DSPのリセットを解除する別の方法は、各コアの $\overline{\text{RS}}$ をハイにしたまま、 $\overline{\text{HPIRST}}$ をローに、次にハイにすることです。これが行われた後、ホストはそれぞれのコアのアドレス0x2Fに任意の値をライトすることによって、対応するコアのリセットを解除することができます。

5

## 5.7 IDLEn時のHPI-16の動作

HPI-16は、特殊なクロック管理回路を利用して、IDLE1状態およびIDLE2の状態の間にも動作できます。この回路は使用するクロックをオンにして同期メモリー・アクセスを実行した後、それらのクロックをオフにして電力を節約します。ホスト・プロセッサは、共用モードのときHPICレジスタのDSPINTビットによってDSPのIDLE状態を解除できます。分離モードでは、外部割り込み(たとえば $\overline{\text{INT0}}$ 、 $\overline{\text{INT1}}$ )を使用してDSPのIDLE状態を解除できます。

ホスト・プロセッサは、HPICレジスタのDSPINTビットに1をライトすることでDSPのIDLE 3を解除できます。IDLE 3時のホスト・プロセッサによるメモリー・アクセスはサポートされていません。

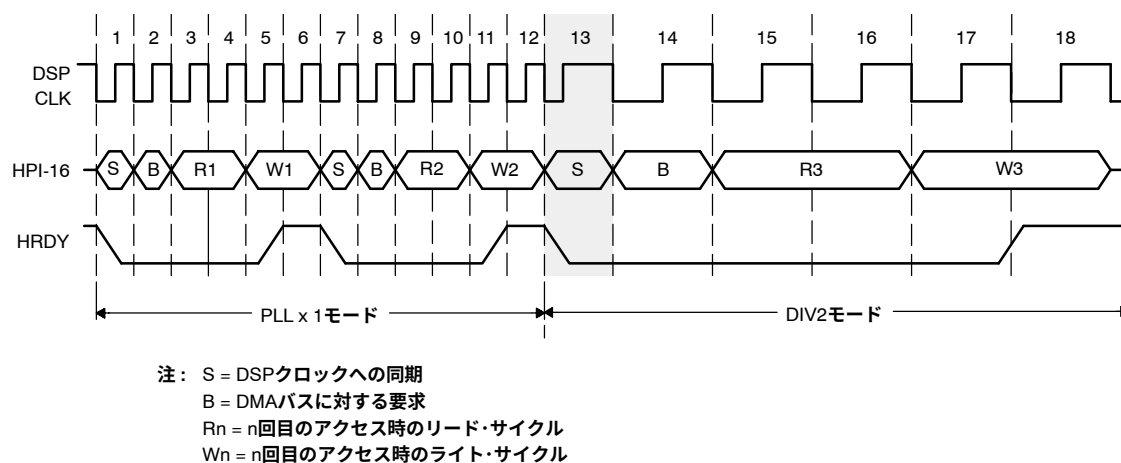
## 5.8 DSPクロック・モードの変更によるHPI-16への影響

前述したように、HPIはDSPクロックに同期します。HPIは6-DSPサイクルごとにDSPの内部メモリーにアクセスできます。したがってDSPのクロック・モードを変更すると、メモリー・アクセスを完了するために必要な時間に影響します。DSPのクロックを遅くすると、HPIのアクセス速度もそれに比例して遅くなります。

例えば、クロック・モードをPLLモードからDIVモードに変更した場合、HRDYピンを使ってホストにウェイトステートを挿入できます。図5-13は、1通倍からDIVモードに変更した場合のタイミングを示しています。HRDYピンを使用することで、ウェイトステートが挿入され、その時点のアクセスが完了するまでホストが待機します。

5

図5-13. PLLからDIVへのクロック・モード変更時のHPI-16の動作



DSPのクロックをn通倍のPLLモードに変更することで、実際のHPIの転送速度が向上します。この場合、必要なホストへのウェイトステートは少なくなります。実際、通倍率を増やすと、HPIの転送速度がホスト・アクセス速度のレイテンシーを超えることもあり得ます。このような場合は、ホストへのウェイトステートは必要ありません。

## 内部でのプロセッサ間通信

この章では、マルチコアDSPの内部でのプロセッサ間通信について説明します。これには、コア間FIFO通信と、外部メモリー・インターフェイスからホスト・ポート・インターフェイス (EMIFからHPI)への通信が含まれます。

| Topic                            | Page |
|----------------------------------|------|
| 6.1 マルチコアDSP内での通信 .....          | 6-2  |
| 6.2 双方向FIFO .....                | 6-3  |
| 6.3 外部メモリー空間からHPI-16へのアクセス ..... | 6-7  |
| 6.4 McBSPを使用するサブシステム通信 .....     | 6-10 |
| 6.5 内部プロセッサ間割り込み .....           | 6-11 |

## 6.1 マルチコアDSP内での通信

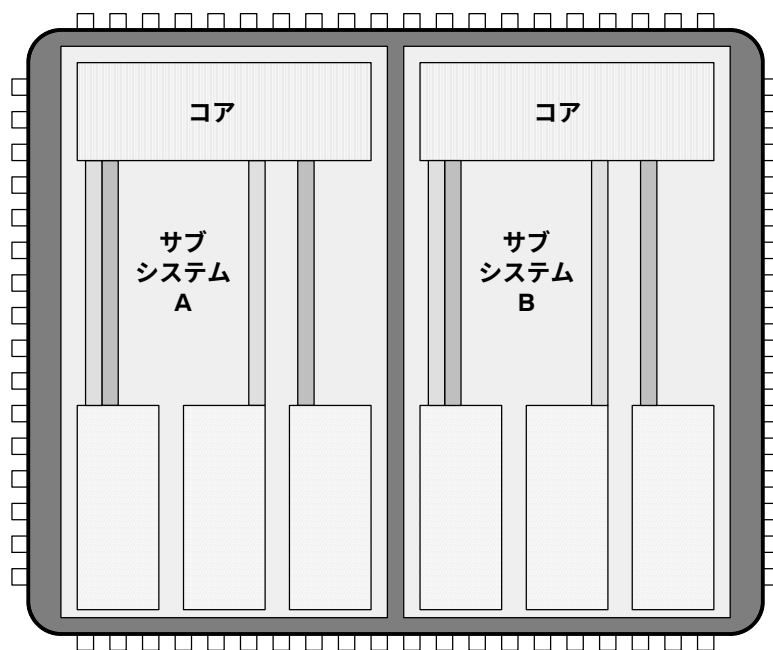
マルチコア(マルチCPU)、マルチサブシステムDSPは、テキサス・インスツルメンツによって1999年に導入されました。これらのデバイスは、複数のコアで同じアプリケーションを実行したり、1つのアプリケーションを各コア間でタスクを分割して実行することにより、パフォーマンスを2倍まで引き上げることができます。サブシステムはそれぞれ独自のコアを備え、互いに独立しています。

複数のコア間でタスクを分割するには、サブシステム間でデータ転送と情報伝達を行う必要があります。プロセッサ間通信を実行するには、次のような方法があります。

- ☐ FIFOの使用
- ☐ ホスト・ポート・インターフェイス(HPI)
- ☐ マルチチャネル・バッファド・シリアル・ポート(McBSP)
- ☐ 相互メモリー空間

この章では、例としてVC5420を使用し、これらのタスク管理方法について説明します。各デバイスの詳細な情報については、データ・シートを参照してください。

図6-1. VC5420 -- 2サブシステムDSP

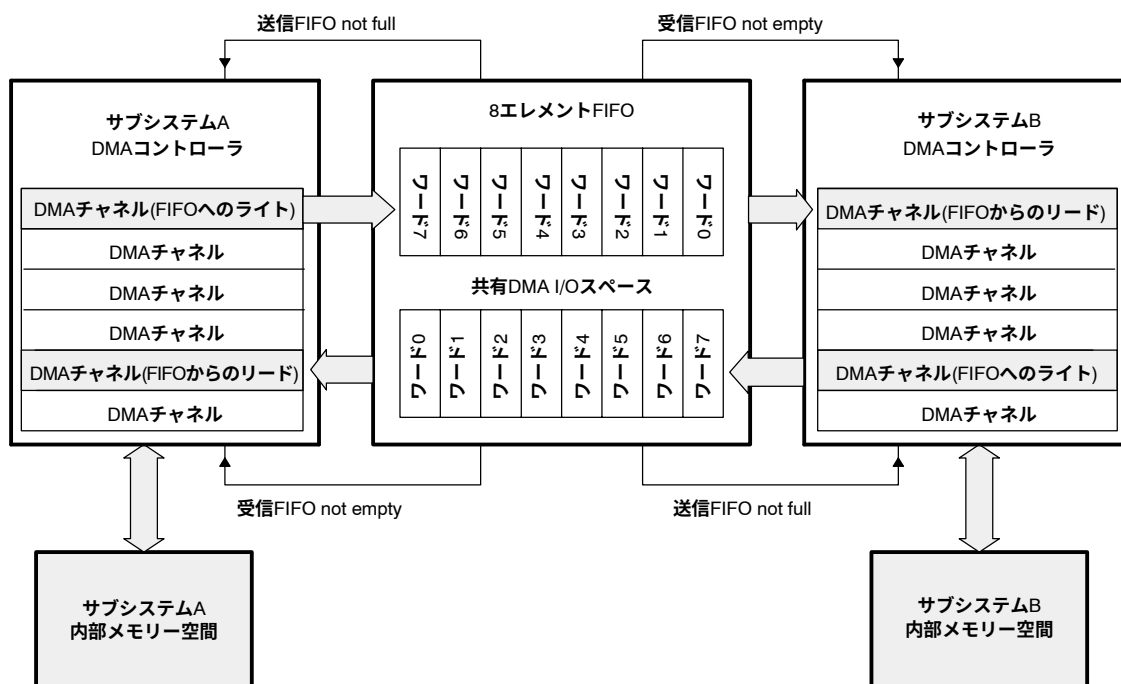


## 6.2 双方向FIFO

サブシステム間の通信は、この節で説明するファーストイン・ファーストアウト(FIFO)方式によって行うことができます。他の方法については、6-10ページの第6.4節で説明します。

FIFOペリフェラルは、単方向、容量8エレメントの2つのチャンネルで構成されます。各チャンネルは16ビット・データ(エレメント)転送をサポートし、送信動作または受信動作専用です。<sup>1</sup>VC5420の場合、FIFOは両方のサブシステムによって共有されます。一方のFIFOチャンネルはAからBへのデータ送信専用であり、他方のチャンネルはBからAへのデータ送信専用です。データを送信しているサブシステムDMAコントローラは、FIFOが一杯になるまで、FIFOに8ワードをライトすることができます。しかし、受信側のサブシステムが同時にその受信チャンネルからリードする場合は、送信側のサブシステムは無限に送信を継続できます。

図6-2. <sup>1</sup>VC5420のFIFO構成



8エレメントFIFOバッファのオーバーラン状態およびアンダーラン状態を処理するために、特殊な内部ハンドシェイク方法が用意されています。そのため、どちらの状態でもデータが失われることはありません。これは、データ転送速度が送信サブシステムと受信サブシステムの両方の動作状況に依存することを意味します。

<sup>1</sup>VC5420では、FIFOとDMAの間でハンドシェイクが実行され、一度DMAの受信チャンネルと送信チャンネルを設定してイネーブルにすればCPUによる介入は必要ありません。例6-1に、DMAチャンネル0および1をそれぞれFIFO送信および受信に設定したDMAチャンネルの情報を示します。

## 双方向FIFO

### 例6-1. DMAチャンネル0および1のFIFO送受信の設定

#### 送信 -- DMAチャンネル0

DMSRC0 = #200h    - 転送元アドレス  
DMDST0 = #xxxh    - 不特定  
DMCTR0 = #0Fh    - エLEMENT・カウンタ(バッファ・サイズ)  
DMSFC0 = #8000h   - DMA同期モード  
DMMCR0 = #0142h   - DMAインクリメント・モード、メモリー空間情報

#### 受信 -- DMAチャンネル1

DMSRC1 = #xxxh    - 不特定  
DMDST1 = #400h    - 転送先アドレス  
DMCTR1 = #0Fh    - エLEMENT・カウンタ(バッファ・サイズ)  
DMSFC1 = #7000h   - DMA同期モード  
DMMCR1 = #085h    - DMAインクリメント・モード、メモリー空間情報

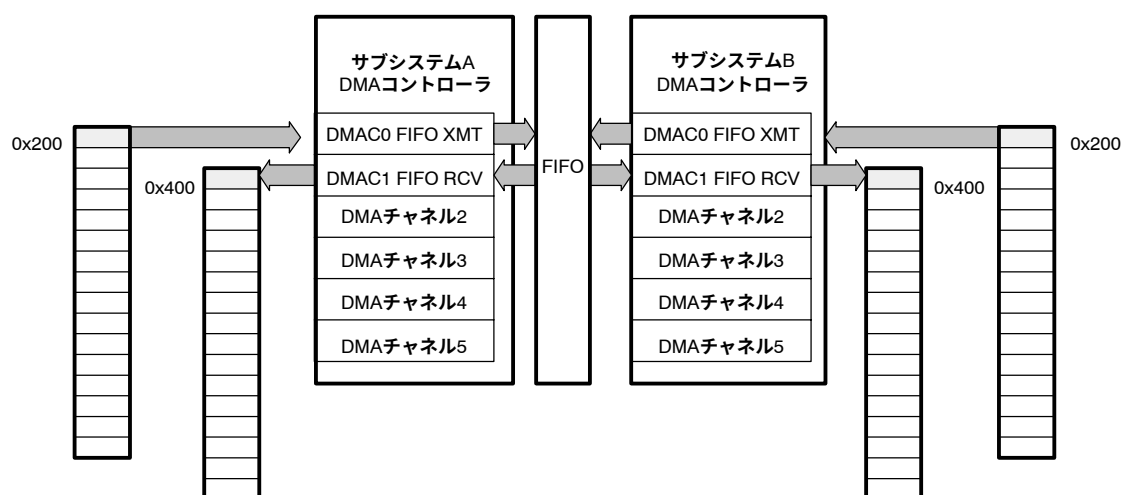
6

表6-1. FIFO転送をサポートするDMA設定

| 送信チャンネル(DMAチャンネル0)                       |                       |           | 受信チャンネル(DMAチャンネル1)                       |                       |            |
|------------------------------------------|-----------------------|-----------|------------------------------------------|-----------------------|------------|
|                                          | 転送元                   | 転送先       |                                          | 転送元                   | 転送先        |
| 転送元および転送先の場所                             | 内部メモリー                | FIFO      | 転送元および転送先の場所                             | FIFO                  | 内部メモリー     |
| メモリー空間                                   | データ/プログラム             | I/O       | メモリー空間                                   | I/O                   | データ/プログラム  |
| メモリー・アドレス                                | 200h                  | 不特定       | メモリー・アドレス                                | 不特定                   | 400h       |
| DMAインデックス・モード<br>(バッファ・ポインタ・インクリメント・モード) | ポストインクリメント            | インクリメントなし | DMAインデックス・モード<br>(バッファ・ポインタ・インクリメント・モード) | インクリメントなし             | ポストインクリメント |
| 内部DMA同期イベント                              | 送信FIFO not full       |           | 内部DMA同期イベント                              | 受信FIFO not empty      |            |
| ELEMENT・カウンタ<br>(バッファ・サイズ)               | 0Fh                   |           | ELEMENT・カウンタ<br>(バッファ・サイズ)               | 0Fh                   |            |
| 割り込み選択                                   | DMA CH0 = IMR/IFRビット6 |           | 割り込み選択                                   | DMA CH1 = IMR/IFRビット7 |            |

図6-3に、表6-1の設定におけるDMAの実際の動作を示します。DMAがCPUの介入なしでデータ・ストリームを処理できるよう、転送バッファ(内部データ空間または内部プログラム空間のどちらかにある)を初期化する必要があります。データ・バッファは合計32ワード(16ワードの受信バッファと16ワードの送信バッファ)で、内部メモリー内の200hおよび400hに配置します。

図6-3. FIFO動作のためのDMA構成



6

FIFOは共有のDMA I/O空間にあります。つまり、両サブシステムのDMAが共通のI/O空間を共有します。そのため、DMAメモリー空間選択はFIFOにアクセスするI/O空間に設定する必要があります。メモリー空間選択はDMMCRレジスタで行います。

リード/ライトはすべて1つのFIFOメモリー位置で行われるため、FIFOを指す転送元/先アドレスは参照されません。したがって、DMAのFIFO側のインデックス・モードは、インクリメントなしに設定する必要があります。バッファ・サイズ(DMCTRレジスタ:エレメント・カウンタ)は、常に(希望のバッファ・サイズ-1)にプログラムします。エレメント・カウンタは転送元バッファと転送先バッファの両方に使用されます。

DMAチャンネル0および1の割り込みは、IMRビット6および7に割り当てられています。しかし、例6-1および表6-1では割り込みをプログラムまたはイネーブルにしています。割り込みをイネーブルにするには、DMAチャンネルをIMRビットに割り当てる必要があります。また、DMAはFIFOへのライト中、特定の時期に割り込みを発生するようプログラムする必要があります。

割り込みは、FIFOバッファが50%または100%一杯になったときに発生させることができます。これらの割り込みは、DMAのDMMCRレジスタを設定し、CPUの割り込みマスク・レジスタ(IMR)でイネーブルにされます。バッファの半分がフル状態で割り込みを発生させている場合は、4番目のワード受信が完了したときに割り込みが発生します。バッファが全てフル状態で割り込みを発生させる場合は、8番目のワード受信が完了した後に発生します。

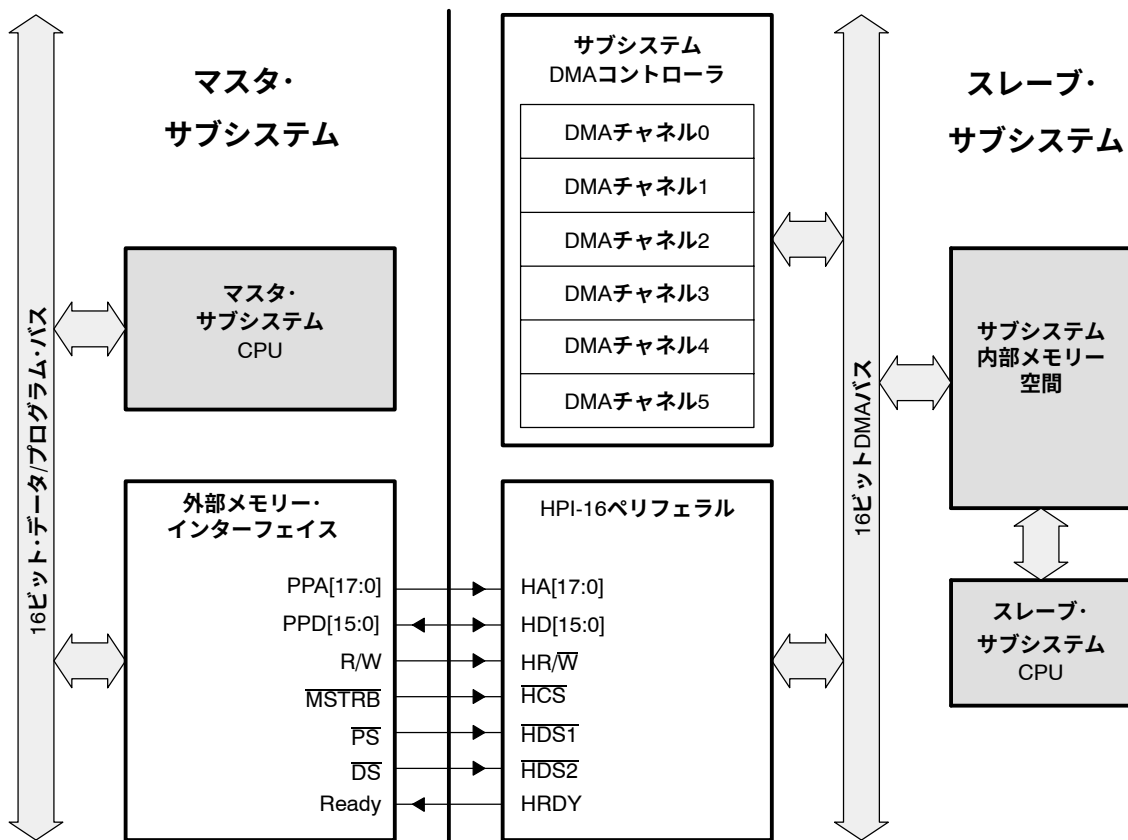
FIFOに1ワードをライトするときのレイテンシーは4CPUサイクルです。受信サブシステムが即座にそのワードをリードする場合は、FIFOからデータをリードするのにさらに4-CPUサイクルが必要です。したがって、最大転送速度は8CPU(16ビット・ワード)です。DMAコントローラが他のDMAチャンネルまたはHPIの転送を処理していると、さらにレイテンシーが発生します。



### 6.3 外部メモリー空間からHPI-16へのアクセス

スタンダードHPI、HPI-8、およびHPI-16ペリフェラルはすべて、別のDSPの外部メモリー空間に接続できるよう設計されています。HPIは常にスレーブ・デバイスであるため、マスタDSPはHPIをストローブするために外部メモリーへのリード/ライトを実行する必要があります。マルチコアDSPでは、この接続が内部で行われます。図6-4に、VC5420における外部メモリー・インターフェイスとHPI-16との内部接続を示します。

図6-4. HPI-16と外部メモリー・インターフェイスの接続(VC5420)



HPI-16は、内部メモリー空間からデータ・アクセスを実行できる16ビット・デバイスです。18ビット・アドレス・バスがあるため、HPI-16は全てのアドレス・レンジをアドレッシングできます。HPI-16は、分離モードで動作し、 $\overline{\text{HDS}}$ ラインをストローブすることで動作します。マスタ・デバイスは外部メモリーにアクセスしなければなりません。外部プログラム/データ・メモリー・インターフェイス信号はHPI-16の制御ピンに接続されています。

## 外部メモリー空間からHPI-16へのアクセス

マルチコアDSPのサブシステムは、HPI-16と外部メモリー・インターフェイスを共有しています。そのため、通信の方向(SELA/Bピンによって)を変更しなければならない場合は、外部ロジックが必要となります。HPI-16は分離モードで動作する必要がある、HMODEピンをハイにする必要があります。EMIFピンが内部でHPI-16ピンに内部で接続される特殊な状態にするために、XIOピンもハイに接続する必要があります。表6-2に、それぞれのピン設定における動作の概要を示します。

表6-2. EMIF/HPI-16モード

| HMODE | SELA/B | EMIFモード(XIO = 1)                            |
|-------|--------|---------------------------------------------|
| 1     | 0      | EMIF-HPIマスタはサブシステムA<br>EMIF-HPIスレーブはサブシステムB |
| 1     | 1      | EMIF-HPIマスタはサブシステムB<br>EMIF-HPIスレーブはサブシステムA |

通信方向を変更する場合のもう1つの条件は、PMSTレジスタのMP/ $\overline{MC}$ ビットの値です。サブシステム間のハンドシェークには、SELA/Bピンの状態変更を要求/応答するだけでなく、両サブシステムのMP/ $\overline{MC}$ ビットの値も変更する必要があります。初期化が必要なのは、リセット後のみです。表6-3に、XIOピン、HMODEピン、およびSELA/Bピンの状態による、リセット時のMP/ $\overline{MC}$ ビット・レベルを示します。

表6-3. リセット時のMP/ $\overline{MC}$ ビット・レベル

| 'C5420ピン |       |        | MP/ $\overline{MC}$ ビット |         |
|----------|-------|--------|-------------------------|---------|
| XIO      | HMODE | SELA/B | サブシステムA                 | サブシステムB |
| 0        | X     | X      | 0                       | 0       |
| 1        | 0     | X      | 1                       | 1       |
| 1        | 1     | 0      | 1                       | 0       |
| 1        | 1     | 1      | 0                       | 1       |

リセット後、ハンドシェーク・ロジックとサブシステムにより、SELA/BピンおよびMP/ $\overline{MC}$ ビット・レベルを設定しなくてはなりません。サブシステムAのMP/ $\overline{MC}$ 値は、常にSELA/Bピンと反対のロジック・レベルに設定する必要があります。さらに、サブシステムBではMP/ $\overline{MC}$ ビットを常にSELA/Bピンと同じロジック・レベルに設定する必要があります。

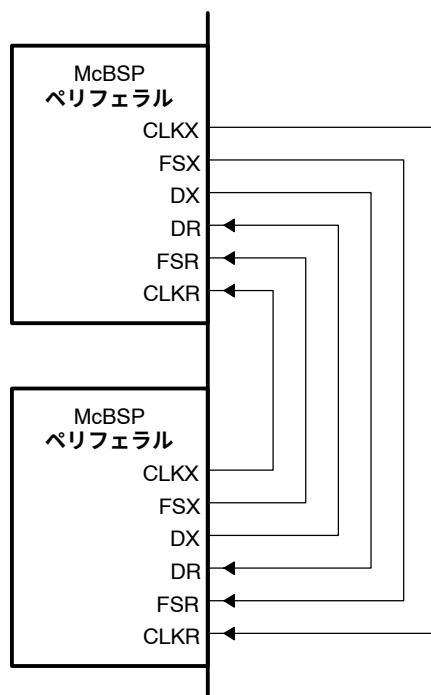
XIOピンとHMODEピンの状態により、マスタ・デバイスは外部メモリーから実行する必要があります。スレーブ・サブシステムはHPIモードで動作するように設定します。したがって、すべての内蔵メモリー・アクセスがイネーブルになります。その結果、マスタ・サブシステムがアクセスする外部メモリー領域は、スレーブ・サブシステムがアクセスする領域と同じになります。

EMIF/HPI-16チャンネルの最大転送速度は、ほかのDMAチャンネルが動作していない場合、6-DSPクロック・サイクルです。別のDMAチャンネルが動作中の場合は、転送速度が14 DSPクロック・サイクルにまで低下することがあります。しかし、HPIレディー・ピン(HRDY)がDSPのCPUのREADYピンに接続されているため、マスタ・デバイスはウエートステートを挿入する必要はありません。

## 6.4 McBSPを使用するサブシステム通信

VC5420マルチコアDSPはMcBSP間通信をサポートしていませんが、各サブシステムのMcBSPを外部で接続することで、非常に柔軟なプロセッサ間通信を実現できます。図6-5に、この外部接続を示します。6本のラインを接続するだけで、双方向/全二重通信チャネルを実現します。マスタ/スレーブ構成にする必要はありません。

図6-5. McBSP間通信



McBSPの送信部は、他方のサブシステムのMcBSP受信部にシフト・クロックとフレーム同期を供給する必要があります。したがって、McBSPのCLKXピンとFSXピンは出力に設定されます。CLKRピンとFSRピンは入力として設定します。

McBSPは多様なデータ転送モードをサポートするようプログラムできます。サブシステムのDMAコントローラを、自動バッファリング送受信モードにして使用できます。さらにMcBSPは、特定の条件でCPU割り込みを発生させるようプログラムできます。

McBSPシリアル・ポートをプロセッサ間通信に使用する方法是、サブシステム通信の最も簡単な形態の1つです。McBSP間の接続は内部ではサポートされません。システム・ボード・レベルでこれを接続して下さい。

## 6.5 内部プロセッサ間割り込み

DMA割り込み、HPI割り込み、およびMcBSP割り込みでは十分でない場合、内部プロセッサ間割り込み(IPINT)を使用できます。この割り込みは内部で発生して、他方のサブシステムに特定のイベントを通知します。この割り込みを発生させるには、CPUがBSCRレジスタの内部プロセッサ間割り込み要求ビット(IPIRQ)に1をライトする必要があります。IPIRQビットは、各サブシステムのBSCRレジスタに割り当てられています。さらに、各サブシステムはそれぞれのIPINT割り込みに対して応答できます。

割り込み元のサブシステムは、IPIRQビットに1をライトし、続いて0をライトする必要があります。その結果、割り込み先のサブシステムのIFRレジスタ(IPINTビット)が更新され、IMRでIPINT割り込みがイネーブルになっていれば、CPUがIPINT割り込みを処理します。割り込み元のサブシステムは、IPIRQビットをハイのままにしておくことができます。これによって割り込み先のサブシステムに複数の割り込みが発生することはありません。IPINT割り込みを表すIFRフラグは、対応する割り込みサービス・ルーチンをCPUが処理したときに、自動的にクリアされます。IPIRQビットをゼロにクリアするまでは、後続の割り込みを発生できません。

内部プロセッサ間割り込みは、すべてのマルチコアDSPでサポートされているわけではありません。各デバイスのデータ・シートを参照し、当該デバイスでこの機能がサポートされているかどうか確認してください。



## 拡張外部バス・インターフェイス(EnhXIO)

この章では、VC5410デバイスで利用できる拡張外部パラレル・インターフェイス(XIO2)の強化された機能について説明します。

| Topic                              | Page |
|------------------------------------|------|
| 7.1 拡張外部バス・インターフェイス(XIO2)の概要 ..... | 7-2  |
| 7.2 バス・タイミング .....                 | 7-3  |

### 7.1 拡張外部バス・インターフェイス(XIO2)の概要

拡張外部パラレル・インターフェイスは、特定の'C54xデバイスでのみ使用できます。  
'VC5410の外部インターフェイス(XIO2)には、次のような改良が加えられています。

- ☐ バス・タイミングの簡素化
- ☐ リード動作からライト動作に、またはその逆に遷移する際のバスの競合回避の強化
- ☐ DMAコントローラへの外部メモリー・アクセス
- ☐ パワーダウン・モードの最適化

'VC5410では、以前の'C54xデバイスと同じインターフェイス信号を用いますが、信号のタイミングが簡素化されています。ほとんどの外部アクセスには、先行サイクル、アクティブ(リードまたはライト)サイクル、および後続サイクルの3サイクルが必要になりました。

先行サイクルと後続サイクルは、リード動作とライト動作の切り替え時のバス競合を回避します。高速リード・アクセスを維持するため、以前の'C54xデバイスと同様、単一サイクルのリードも使用できます。

#### 7.1.1 追加機能

XIO2は、DMA転送を外部メモリーまで拡張する機能も備えています。DMAの機能の詳細については、第3章を参照してください。

XIO2により、インターフェイスに内部クロックが使用されていないときにこれをオフにすることで、'C54xデバイスが既に備えている低消費電力性能を高めています。この節電機能は自動で、ソフトウェア・セットアップが必要なく、インターフェイスの動作にレイテンシーを発生しません。

XIO2には、更に次のような追加機能があります。

- ☐ 32Kのメモリー境界をアクセスがまたぐときにバンク・スイッチング・サイクルを自動挿入する機能
- ☐ ソフトウェアにより最大14のウェイトステートを挿入する機能
- ☐ CLKOUTを1、2、3、または4分周する機能。CLKOUTを分周することで、低速の外部メモリーまたはペリフェラル・デバイスとのインターフェイスにおいて、ウェイトステートを挿入する以外の方法を提供します。

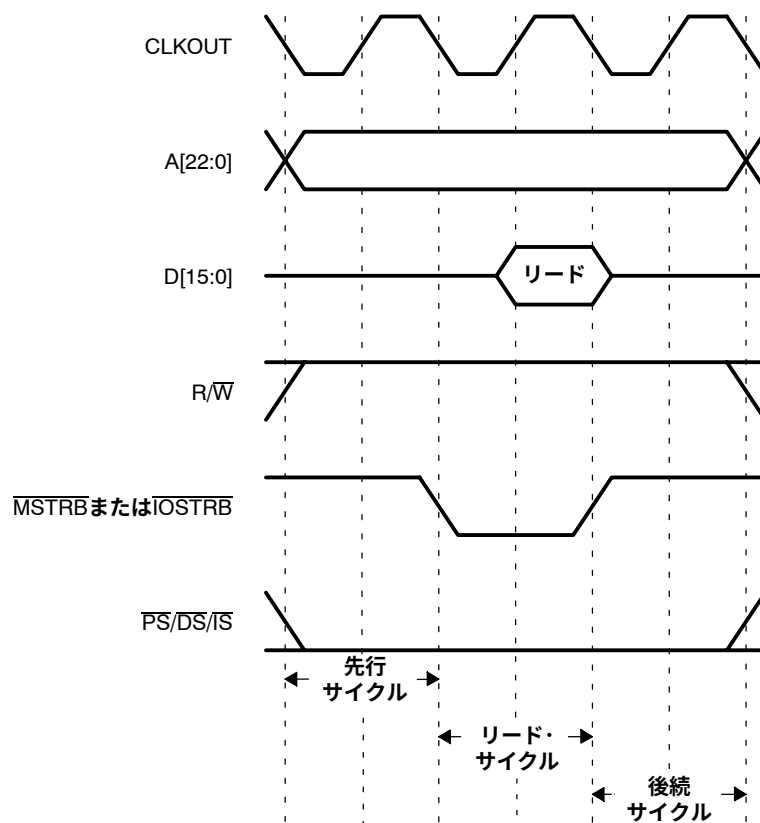
ウェイトステートを挿入すると、リード・アクセスまたはライト・アクセス中のバス・シーケンスが長くなりますが、アクセス開始時および終了時のバス信号のタイミングは遅くなりません。CLKOUTを分周することで、必要な場合にバスのタイミング全体を遅くする方法をとることができます。CLKOUTの分周率は、バンク・スイッチング制御レジスタ(BSCR)のDIVFCTフィールドによって制御されます。



## 7.2 バス・タイミング

図7-1に、すべてのI/Oリード、不連続モードでのメモリー・リード、および連続モードでの1回のメモリー・リードという3つの場合におけるバス・タイミングを示します。図中のアクセスが完了するには、常に3CLKOUTサイクルが必要です。

図7-1. 不連続メモリー・リードおよびI/Oリードのバス・タイミング



## バス・タイミング

図7-2に、連続モードでメモリー・リードを繰り返した場合のバス・タイミングを示します。  
図中のアクセスが完了するには、 $(2+n)$  CLKOUTサイクルが必要です。ここで $n$ は、連続  
リードの実行回数です。

図7-2. 連続メモリー・リードのバス・タイミング( $n = 3$  リード)

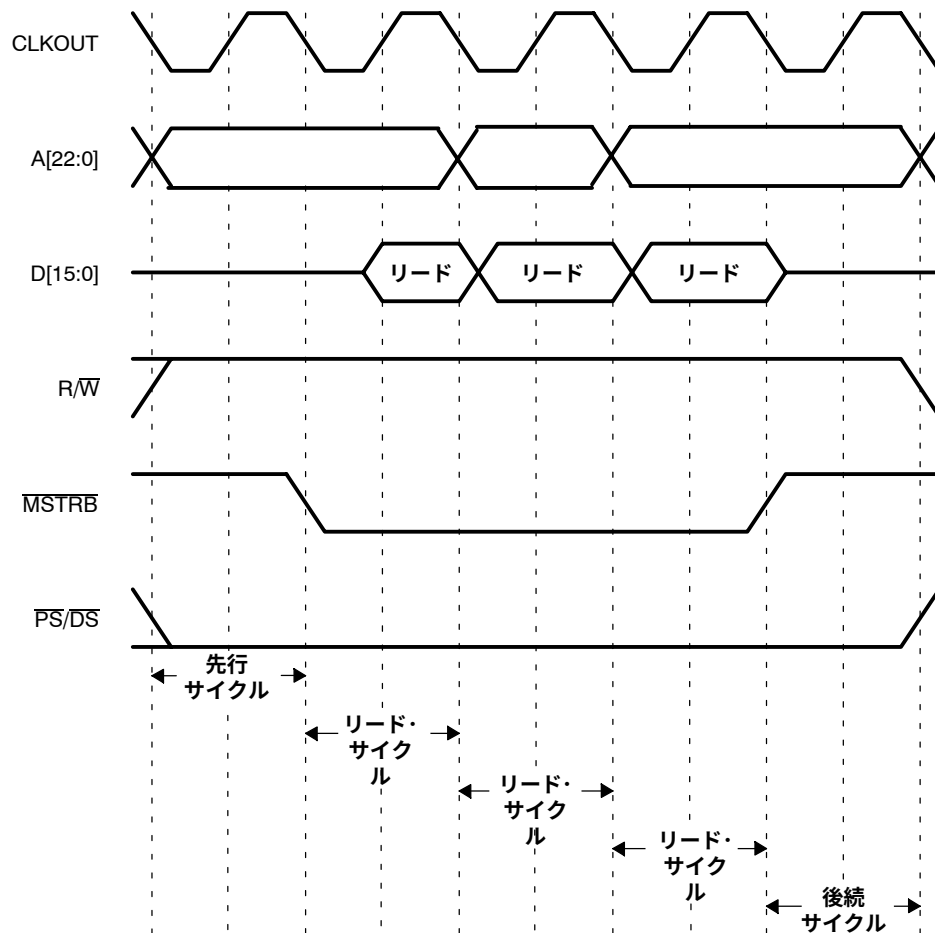
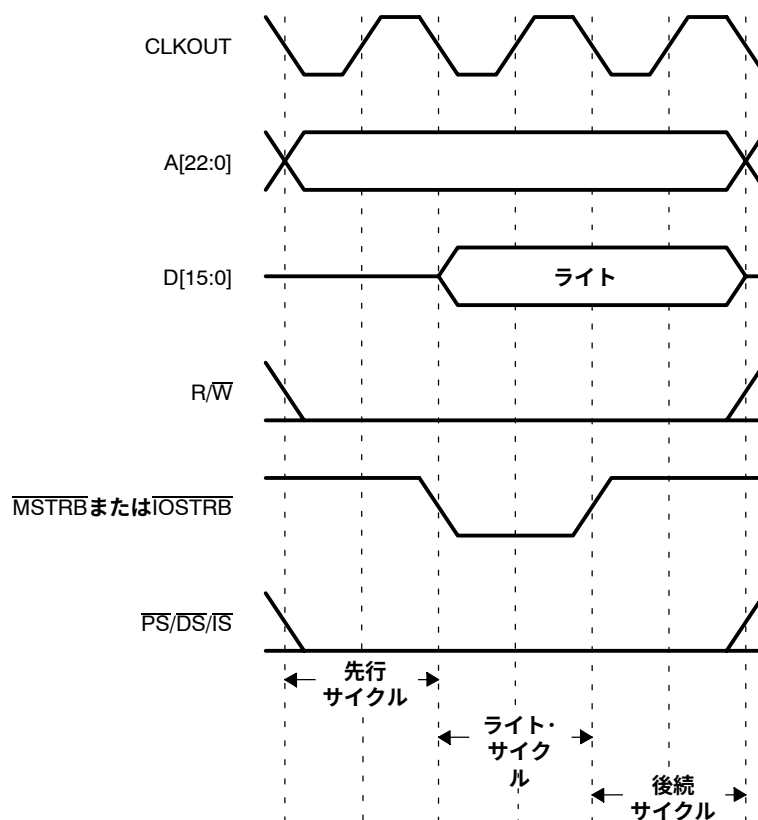


図7-3に、すべてのメモリー・ライトおよびI/Oライトのバス・タイミングを示します。図中のアクセスが完了するには、常に3CLKOUTサイクルが必要です。

図7-3. メモリー・ライトおよびI/Oライトのバス・タイミング



\_\_\_\_\_

## 用語集

## A

**A-bisモード** A-bisモード (ABIS=1)では、McBSPがPCMリンク上で最大1024ビットのデータを送受信できる。

**AC97** Audio Codec '97。デュアル・フェーズ・フレーム機能を使用する規格。第1フェーズは1つの16ビット・ワードで構成され、第2フェーズは12個の20ビット・ワードで構成される。

**ALE** アドレス・ラッチ・イネーブル。アドレス/データ・バス共用型ホストでは、このALEを使ってアドレス/制御データをラッチする場合がある。通常は $\overline{\text{HAS}}$ に接続される。

**AUTOINIT** DMA自動初期化ビット。DMAのグローバル・レジスタからロードすることで、フレーム後に自動的に初期化を行うようチャンネル設定する。AUTOINITはDMMCRレジスタ上に割り当てられている。

## A

## B

**BOB** HPIバイト・オーダー・ビット。最初に転送されるバイトを選択するビット。HPI-8およびスタンダードHPIでのみ使われる。BOBはHPICレジスタ上に割り当てられている。

## C

**CLKG** プログラマブル・データ・クロック。受信/送信のクロックとフレームを生成する場合に、プログラムできるMcBSPの内部信号。

**CLKGDV** サンプル・レート生成器に対する入力クロック・ソースを分周する場合に使用できる、プログラム可能な値。

**CLKR** 受信クロック。McBSPのインターフェイス信号。

**CLKRM** 受信クロック・モード。CLKRMはPCRレジスタ上に割り当てられている。

**CLKRP** 受信フレーム・クロック極性ビット。CLKRPはPCRレジスタ上に割り当てられている。

**CLK(R/X)** データ・クロック(受信/送信)。

**CLKS** 外部クロック入力。McBSPのインターフェイス信号。

**CLKSM** クロック・ソース・モード。SRGR2内のCLKSMビットは、CPUクロック (CLKSM=1)または外部クロック入力(CLKSM=0)CLKSを、サンプル・レート生成器の入力クロックに対するソースとして選択する。

**CLKSP** クロック極性。CLKSP=0の場合はCLKSの立ち上がりエッジ、CLKSP=1の場合はCLKSの立ち下がりエッジにより、データ・ビットレート・クロック(CLKG)とフレーム同期(FSG)が変化する。

**CLKSRG** クロック・サンプル・レート生成器。CLKSRGの立ち上がりエッジにより、クロック(CLKG)とフレーム(FSG)が生成される。

**CLKS\_STAT** CLKSピン・ステータス・ビット。CLKSが汎用入力として設定されている場合に、CLKSピンのロジック・レベルを示す。

**CLKSTP** クロック・ストップ・モード・ビット。CLKSTPはSPCR1レジスタ上に割り当てられていて、シリアル・ポート・クロックを停止するのに使われる。

**CLKX** 送信クロック。McBSPのインターフェイス信号。

**CLKXM** 送信クロック・モード。CLKXMはPCRレジスタ上に割り当てられている。

**CLKXP** 送信クロック極性ビット。CLKXPはPCRレジスタ上に割り当てられている。

**CTMOD** DMA転送カウンタ・モード制御ビット。マルチフレーム・モードまたはABUモードを選択する。CTMODはDMMCRレジスタ上に割り当てられている。

## D

**DATDLY** データ遅延(受信/送信)。実際のデータ受信/送信の開始時に発生するフレーム先頭からの遅延。プログラム可能なデータ遅延の範囲は0~2ビットクロック。

**DBLW** ダブルワード・モード。このモードはDMAデータのサイズが32ビットの場合に使われる。DBLWはDMSFCレジスタ上に割り当てられている。

**DE** DMAチャネル・イネーブル・ビット。DMAのチャネル動作のイネーブル/ディセーブルを切り替える。DEはDMPRECレジスタ上に割り当てられている。

**DIND** DMA転送先・アドレス・インデックス・モード・ビット。このビットでは、数種類のインデックス・モードが用意されている。DINDはDMMCRレジスタ上に割り当てられている。

**DINM** DMA割り込み発生マスク・ビット。DMA転送割り込みのイネーブル/ディセーブルを切り替える。DINMはDMMCRレジスタ上に割り当てられている。

**DIR** 方向ビット。汎用I/Oピンを入力または出力に設定する。DIRはGPIOCRレジスタ上に割り当てられている。

**DLB** デジタル・ループバック・モード。DLBモードでは、マルチプレクサを介してDR、FSR、CLKRをそれぞれDX、FSX、CLKXに内部連結することにより、1つのDSPデバイスでシリアル・ポートのプログラムをテストできる。

**DMA** ダイレクト・メモリー・アクセス(コントローラ)

**DMCTR** DMAチャンネル・エレメント・カウント・レジスタ

**DMD** DMA転送先アドレス空間選択ビット。DMDはDMMCRレジスタ上に割り当てられている。

**DMDST** DMAチャンネル転送先アドレス・レジスタ

**DMDSTP** DMA転送先プログラム・ページ・アドレス(全チャンネル)

**DMFRI0** DMAフレーム・アドレス・インデックス・レジスタ0

**DMFRI1** DMAフレーム・アドレス・インデックス・レジスタ1

**DMGCR** DMAグローバル・エレメント・カウント・リロード・レジスタ

**DMGDA** DMAグローバル転送先アドレス・リロード・レジスタ

**DMGFR** DMAグローバル・フレーム・カウント・リロード・レジスタ

**DMGSA** DMAグローバル転送元アドレス・リロード・レジスタ

**DMIDX0** DMAエレメント・アドレス・インデックス・レジスタ0

**DMIDX1** DMAエレメント・アドレス・インデックス・レジスタ1

**DMMCR** DMAチャンネル転送モード制御レジスタ

**DMPREC** DMA優先順位およびイネーブル制御レジスタ

**DMS** DMA転送元アドレス空間選択ビット。DMSはDMMCRレジスタ上に割り当てられている。

**DMSA** DMAのサブアドレス・レジスタ用のアドレス・レジスタ。

**DMSDI** DMAサブアドレス・データ・レジスタ(自動インクリメントあり)

**DMSDN** DMAサブアドレス・データ・レジスタ(自動インクリメントなし)

**DMSFC** DMAチャンネル同期イベントおよびフレーム・カウント・レジスタ

**DMSRC** DMAチャンネル転送元アドレス・レジスタ

**DMSRCP** DMA転送元プログラム・ページ・アドレス(全チャンネル)

**DPRC** DMAチャンネルの優先順位を設定する。DPRCはDMPRECレジスタ上に割り当てられている。

- DR** データ受信。McBSPとインターフェイスされているデバイスからデータを受信する。
- DRR[1,2]** データ受信レジスタ1および2。McBSPを介したデータ受信に使われる、2つの16ビットレジスタ。
- DSPINT** ホスト・ポート・インターフェイスのDSP CPU割り込み。ホストでこれを設定して、DSPに割り込みを発生させることができる。DSPINTはHPICレジスタ上に割り当てられている。
- DSYN** DMA同期イベント制御ビット。特定のイベントと同期するようDMAを設定する。DSYNはDMSFCレジスタ上に割り当てられている。
- DX** データ送信。McBSPとインターフェイスされているデバイスにデータを送信する。
- DXENA** データ送信遅延ビット。このビットが設定されている場合は、送信データが遅延処理される。
- DXR[1,2]** データ送信レジスタ1および2。McBSPを介したデータ送信に使われる、2つの16ビットレジスタ。
- DX\_STAT** DXピン・ステータス・ビット。DXが汎用入力として設定されている場合に、DXのロジック・レベルを示す。

## A

## F

- FETCH** 現在のHPIAアドレスにあるデータをフェッチするのに使われるHPIC中のビット。
- FIFO** ファーストイン・ファーストアウト。いくつかのエレメントを順番に格納できるハードウェア・メカニズム。最初にFIFOにライトされたエレメントから先にリードされる。
- FPER** フレーム周期レジスタ。FPERによって、次のフレーム同期信号がアクティブになるタイミングが決定される。
- FREE** フリー・ランニング・モード。FREE = 0の場合はディセーブル、FREE = 1の場合はイネーブル。
- FRLEN (R/X)[1,2]** フレーム長。1フレームあたり転送されるシリアル・ワード(8、12、16、20、24、32ビット)の数。この長さは、フレーム同期信号1つあたりのワード数、論理タイム・スロット数、またはチャンネル数に相当する。
- FRST** フレーム同期生成器のリセット。 $\overline{\text{FRST}}$  = 0の場合は、フレーム同期ロジックがリセットされる。 $\overline{\text{FRST}}$  = 1の場合は、(FPER + 1)回のCLKGクロックをカウント後にフレーム同期信号FSGが生成される。
- FSG** フレーム同期信号。 $\overline{\text{FRST}}$ ビットでアクティブになったフレーム同期生成ロジックによって生成される。



**FSGM** サンプル・レート生成器の転送フレーム同期モード・ビット。FSGMはSRGR2レジスタ上に割り当てられている。

**FSR** 受信フレーム同期。McBSPのインターフェイス信号。

**FSRM** 受信フレーム同期モード・ビット。FSRMはPCRレジスタ上に割り当てられている。

**FSRP** 受信フレーム同期極性ビット。FSRPはPCRレジスタ上に割り当てられている。

**FSX** 送信フレーム同期。McBSPのインターフェイス信号。

**FSXM** 送信フレーム同期モード・ビット。FSXMはPCRレジスタ上に割り当てられている。

**FSXP** 送信フレーム同期極性ビット。FSXPはPCRレジスタ上に割り当てられている。

**FWID** フレーム幅。フレーム同期パルスのアクティブ幅を制御する、8ビットのダウン・カウンタ。

**G**

**$\overline{\text{GRST}}$**  サンプル・レート生成器のリセット。 $\overline{\text{GRST}} = 0$ の場合は、SPCR2の $\overline{\text{GRST}}$ ビットによって生成器がリセットされる。 $\overline{\text{GRST}} = 1$ の場合は、リセットが解除される。

**GSYNC** サンプル・レート生成器のクロック同期。

**A****H**

**HINT** DSPからホストへの割り込み。DSPはこのビットにライトを行うことで、ホストに割り込みを発生させることができる。HINTはHPICレジスタ上に割り当てられている。

**HMODE** HPIモード。HPI-16でのみ有効。HPIが共用/分離モードで動作できるようにする。

**HPIA** ホスト・ポート・アドレス・レジスタ。アクセスするデータのアドレスが格納される。このレジスタへアクセスできるのはホストに限られる。

**HPIC** ホスト・ポート制御レジスタ。ホストとDSPで使われるレジスタ。

**HPID** ホスト・ポート・データ・レジスタ。このレジスタへアクセスできるのはホストに限られる。HPIを使用すると、データがこのレジスタを介して渡される。

**HPIENA** HPIイネーブル・ステータス・ビット。ホストがHPIENAピンのロジック・レベルを決定できる。HPIENAはHPICレジスタ上に割り当てられている。

**HPIインクリメント・モード** 転送が完了すると、HPIアドレス・レジスタが自動的に修飾されるモード。

**HRDY** ソフトウェアによるHRDYピンのポーリングに使われるHPICレジスタ中のビット (HPI-16のみ)。

**HSTRB** HPIペリフェラルを制御する内部ストローブ。内部のHSTRB信号は、 $\overline{\text{HDS1}}$ 、 $\overline{\text{HDS2}}$ 、 $\overline{\text{HCS}}$ の各外部ピンの状態によって決定される。

## I

**IMOD** DMA割り込み発生モード・ビット。フル/ハーフ・バッファの境界で割り込みが発生するように設定する。IMODはDMMCRレジスタ上に割り当てられている。

**INTOSEL** 割り込み多重制御ビット。DMA割り込みと対応するIFRビットを決定する。INTOSELはDMPRECレジスタ上に割り当てられている。

**IPINT** 内部でのプロセッサ間割り込み。IPINTはIFRレジスタとIMRレジスタ上に割り当てられている。

**IPIRQ** 内部でのプロセッサ間割り込み要求。マルチコアDSP上のサブシステム間で割り込みを発生させるのに使われる。

## A

## L

**LSB** 最下位ビット。ワード内で最も順位の低いビットのこと。

## M

**MSB** 最上位ビット。ワード内で最も順位の高いビットのこと。

## R

**RBR[1,2]** 受信バッファ・レジスタ1および2。受信シフト・レジスタ(RSR)からコピーされたデータを受信する。

**RCBLK** 受信カレント・ブロック。

**RCER[A,B]** パーティションAとパーティションBのMcBSP受信チャネル・イネーブル・レジスタ。

**RCOMPAND** 受信伸長モード。受信データを伸張するようにMcBSPを設定する。

**RDATDLY** 受信データ遅延ビット。0、1、または2ビットのデータ受信遅延を挿入できる。

**RESET** レジスタと制御ビットをあらかじめ定義された値に設定する事でCPUを既知の状態にし、特定のアドレスから実行を開始するよう通知すること。

**REVT** DMAへの受信同期イベント。

**REVT A** A-bisモードにおけるDMAへの受信同期イベント。

**RFIG** 受信フレーム無視ビット。最初のFSRのみ受け入れ、それ以降のFSRは無視する。

**RFRLLEN1** 受信フレーム長1。第1フェーズの受信フレーム長を設定する。

**RFRLLEN2** 受信フレーム長2。第2フェーズの受信フレーム長を設定する。

**RFULL** 受信時のオーバーラン。受信部でオーバーランが発生し、エラー条件になっていることを示す。

**RINT** CPUに対する受信割り込み。

**RIOEN** 受信セクションの汎用I/Oモード。DR、FSR、CLKRの各ピンを汎用I/Oに設定できる。

**RJUST** 受信データの格納方法および符号拡張。SPCR1のRJUSTは、RBR(1,2)のデータがDRR(1,2)内で(MSBに関して)右詰めと左詰めのどちらになるかを選択する。右詰めが選択されている場合はさらに、そのデータが符号拡張されるか、あるいはゼロ詰めされるかを選択する。

**RMCM** 受信マルチチャネル選択イネーブル。

**RPABLK** 受信パーティションAブロック。チャンネルのどのブロックを受信するかを選択する。RPABLKはMCR1レジスタ上に割り当てられている。

**RPBBLK** 受信パーティションBブロック。チャンネルのどのブロックを受信するかを選択する。RPBBLKはMCR1レジスタ上に割り当てられている。

**RPHASE** 受信フェーズ・ビット。McBSPの1フレーム中のフェーズ数を1つ、または2つを選択できる。

**RRDY** 受信レディー・ビット。McBSP受信部がレディー状態であることを示す。

**RRST** 受信リセット・ビット。

**RSR[1,2]** 受信シフト・レジスタ1および2。適切なデータ遅延後に、DRピンからシフトされたデータを受信する。

**rsvd** 予約

**RSYNCERR** 予定外の受信フレーム同期。前のフレームの最終受信ビットの前にRDATDLYビット・クロックよりも早く発生する同期パルス。RSYNCEER=0の場合は同期エラーが検出されていない事を示し、RSYNCERR=1の場合はMcBSPが同期エラーを検出した事を示す。

**RWDLEN1** 受信ワード長1。第1フェーズの受信ワード長を設定する。

A

**RWDLEN2** 受信ワード長2。第2フェーズの受信ワード長を設定する。

**S**

**SIND** DMA転送元・アドレス・インデックス・モード・ビット。SINDには、数種類のインデックス・モードが用意されている。SINDはDMMCRレジスタ上に割り当てられている。

**SOFT** ソフト・ビット。SOFT = 0の場合はSOFTモードがディセーブルになり、SOFT = 1の場合はイネーブルになる。

**SPCR[1,2]** シリアル・ポート制御レジスタ1および2。

**SPSA** McBSPのサブアドレス・レジスタに使われるアドレス・レジスタ。

**SRGR[1,2]** サンプル・レート生成器のレジスタ1および2。

**T**

**TDMデータ・ストリーム** McBSPをシングル・フェーズ・フレームに設定することで、複数チャンネルを送信部と受信部用に独立して選択することができる。各フレームは、時分割多重方式(TDM)のデータ・ストリームを表す。

**A****W**

**WDLEN(R/X)[1,2]** ワード長。送信/受信制御レジスタ内にある3ビットのフィールド。これらのフィールドによって、フレームの各フェーズに対する受信部と送信部のワード長(1ワードあたりのビット数)が決まる。

**X**

**XCBLK** 送信カレント・ブロック。

**XCER[A,B]** パーティションAとパーティションBのMcBSP送信チャンネル・イネーブル・レジスタ。

**XCOMPAND** 送信圧縮モード。送信データを圧縮するようMcBSPを設定する。

**XDATDLY** 送信データ遅延ビット。0、1、または2ビットのデータ送信遅延を挿入できる。

**$\overline{\text{XEMPTY}}$**  送信シフト・レジスタ(XSR[1,2])エンプティ。送信部でアンダーフローが発生した場合に、そのことを知らせる。XSR[1,2]が空の場合は $\overline{\text{XEMPTY}} = 0$ になり、XSR[1,2]が空でない場合は $\overline{\text{XEMPTY}} = 1$ になる。

**XEVT** DMAへの送信同期イベント。

- XEVTA** A-bisモードにおけるDMAへの送信同期イベント。
- XFIG** 送信フレーム無視ビット。最初のFSXのみを受け入れ、それ以降のFSXは無視する。
- XFRLEN1** 送信フレーム長1。第1フェーズに対する送信フレーム長を設定する。
- XFRLEN2** 送信フレーム長2。第2フェーズに対する送信フレーム長を設定する。
- XHPIA** 拡張HPIA。HPIAレジスタから、下位の16ビットと最上位ビットのどちらにアクセスするかを選択する。XHPIAはHPICレジスタ上に割り当てられている。
- XINT** CPUに対する送信割り込み。
- XINTM** 受信/送信割り込みモード。
- XIOEN** 送信セクションの汎用I/Oモード。DX、FSX、CLKXの各ピンを汎用I/Oに設定できる。
- XMCM** 送信マルチチャネル選択イネーブル。
- XPABLK** 送信パーティションAブロック。チャンネルのどのブロックを送信するかを選択する。XPABLKはMCR1レジスタ上に割り当てられている。
- XPBBLK** 送信パーティションBブロック。チャンネルのどのブロックを送信するかを選択する。XPBBLKはMCR1レジスタ上に割り当てられている。
- XPHASE** 送信フェーズ・ビット。McBSPの1フレーム中のフェーズ数を1つまたは、2つを選択できる。
- XRDY** 送信レディー。McBSP送信部がレディー状態であることを示す。XRDY = 0の場合は、送信部がレディー状態になっていない。一方、XRDY = 1の場合は、DXR[1,2]にデータがあり、送信部がレディー状態になっている。
- $\overline{\text{XRST}}$**  送信部のリセット。シリアル・ポート送信部のリセットとイネーブルを行う。  
 $\overline{\text{XRST}} = 0$ の場合は、送信部をディセーブルにし、リセット状態にする。一方  
 $\overline{\text{XRST}} = 1$ の場合は、送信部をイネーブルにする。
- XSR[1,2]** 送信シフト・レジスタ1および2。DXR[1,2]にライトされたデータは、送信シフト・レジスタ(XSR[1,2])を介してDXにシフト・アウトされる。
- XSYNCERR** 予定外の送信フレーム同期(送信同期エラー)。前のフレームの最終送信ビットの前に、XDATDLYビットクロックよりも早く発生する同期パルス。  
XSYNCERR = 0の場合は同期エラーが検出されていないことを示し、  
XSYNCERR = 1の場合はMcBSPがエラーを検出したことを示す。
- XWDLEN1** 送信ワード長1。第1フェーズの送信ワード長を設定する。
- XWDLEN2** 送信ワード長2。第2フェーズの送信ワード長を設定する。

## あ

**アクセス・モード** HPIのアクセス・モードは、HCNTL0/1ピンのロジック・レベルによって決定されます。アクセス・モードはHPICアクセス、HPIAアクセス、HPIDアクセスのいずれかに設定でき、自動インクリメントが行われる。

**圧伸 圧縮と伸張** 音声信号の量子化方法。この方法では、入力信号が圧縮され、処理された後、伸張によって出力時に復元される。圧伸方法には、ヨーロッパで使われているA則と、米国および日本で使われている $\mu$ 則の2種類がある。

**エレメント** 要素。DMAにて転送される基本単位(1、2ワード)。

## か

**拡張プログラマブル・ウェイトステート発生器** 低速のオフチップ・メモリーやI/Oデバイスとのインターフェイスをとるために、外部バスのサイクルに最大14マシン・サイクルまでウェイトステートを挿入できる。

**共用モード** HPIがアドレス/データ・バス共用で動作するモード。

## A

## さ

**サブシステム** マルチコアDSP内のCPU、メモリー、ペリフェラルで構成された独立システムのこと。

**サンプル・レート生成器** サンプル・レート生成器は、プログラム可能なデータ・クロック(CLKG)とフレーム信号(FSG)を生成する3段のクロック分周器で構成される。CLKGとFSGは、受信/送信のクロック(CLKR/X)生成とフレーム(FSR/X)生成を行うためにプログラム可能な、McBSPの内部信号である。サンプル・レート生成器は、内部クロック・ソースか、外部クロック・ソースから供給される内部クロックによって駆動するようプログラムすることができる。

**シリアル・ポート制御レジスタ(SPCR[1,2])** シリアル・ポート・インターフェイス用のステータス・ビットと制御ビットが含まれたレジスタ。

**受信バッファ・レジスタ(RBR[1,2])** RBR[1,2]を参照。

**スレーブ** 別のデバイスによって制御されるデバイス。

**送受信制御レジスタ(RCR[1,2]およびXCR[1,2])** 受信処理と送信処理の各種パラメータを設定する。

## た

**ダイレクト・メモリー・アクセス(DMA)コントローラ** DMAコントローラは、メモリー・マップ上の領域内におけるデータ転送をCPUとは独立に実行します。DMAを使用すれば、内部メモリー、内部ペリフェラル、または外部デバイス間のデータ転送をCPU動作のバックグラウンドで行うことができます。

## は

**汎用入力/出力ピン(GPIO)** 外部デバイスから入力信号を受けたり、外部デバイスへ信号を出力するのに使われるピン。これらのピンは特定の用途に使われるのではなく、幅広い目的の入力/出力信号の扱いができる。GPIOには、汎用BIO入力ピンやXF出力ピンなどがある。

**ビット・オーダーリング** 圧伸データが使われていない場合に、LSBからシリアル・ポートに転送できる機能。

**ピン制御レジスタ(PCR)** 通常のシリアル・ポート動作中のMcBSPピンを、入力/出力として設定するのに使われる。また、受信部/送信部のリセット時に、シリアル・ポート・ピンを汎用入力/出力として設定するのにも使われる。

**フレーム・カウント** DMAフレーム・カウント。転送されるフレーム数を指定する。フレーム・カウントはDMSFCレジスタ上に割り当てられている。

**プログラマブル・ウェイトステート発生器** 「拡張プログラマブル・ウェイトステート発生器」を参照。

**プログラマブル・バンクスイッチング・モジュール** ターン・オフに数サイクル必要なメモリーを'C54xが外部ウェイトステートを要求することなく、外部メモリーバンク間を切り替えることができます。プログラムまたはデータ空間内で32Kワードのメモリーバンク境界を超えてアクセスすると、バンクスイッチング・ロジックにより自動的にサイクルが1つ挿入される。

**分離モード** HPIが個別のアドレスおよびデータ・バスを使って動作するモード。

**ホスト** HPIペリフェラルを駆動する外部デバイス。

**ホスト・ポート・インターフェイス(HPI-8)** ホスト・デバイスまたはホスト・プロセッサと'C54xとのインターフェイスとなる拡張型の8ビット・パラレル・ポート。'C54xとホスト・デバイス間の情報交換のために、'C54xの全てのオンチップRAMが使用可能である。

**ホスト・ポート・インターフェイス(HPI-16)** 'C54x 8ビット・ホスト・ポート・インターフェイスの拡張型16ビット・バージョン。HPI-16は完全な16ビット双方向データ・バスを搭載しており、バイト識別を必要としない。内部メモリーのアドレス範囲全域にアクセスすることができる。

A

## ま

**マスタ** ペリフェラルやバスなどを制御するデバイス。

**マルチチャネル・バッファド・シリアル・ポート(McBSP)** McBSPは高速かつ全二重双方向のマルチチャネル・バッファド・シリアル・ポートで、他の'54xデバイス、コーデック、およびシステム内の他のデバイスと直接インターフェイスを取ることができる。さらに、McBSPはダブルバッファド・レジスタを搭載しているため、連続したデータ・ストリームと、送受信時の独立したフレーム処理やクロック処理を可能にしている。

## れ

**レイテンシー** ある条件が発生してから、デバイスがその条件に反応するまでの遅延時間。  
また、パイプライン中の2つの命令実行の間に、2番目の命令が使用する値が正しくなるための必要な遅延の事。



# 索引

## A

A-bis インターフェイス機能, 2-83  
A-bis モード  
  受信動作, 2-84  
  送信動作, 2-84  
  定義, A-1  
ABIS モード, 2-8  
ABU のアドレス・インデックス・モード, 3-25  
ABU バッファ・サイズの例, 3-25  
ABU バッファの例, 3-24  
ABU モードでのハーフバッファ割り込みの発生  
  'C5402/'C5420, 3-28  
  'C5410, 3-31  
ABU(自動バッファリング)モード, 3-23  
AC97  
  定義, A-1  
  デュアル・フェーズ・フレーム・フォーマット,  
    2-36  
  複数フェーズ・フレームの例, 2-35  
  フレーム同期付近のビット・タイミングの例,  
    2-36  
ALE, 定義, A-1  
AUTOINIT, 定義, A-1

## B

BCLKX 信号, 2-92  
BOB, 定義, A-1  
BOB と HPIA の初期化, 4-20  
BSP シリアル・ポート制御拡張レジスタ(SPEC)  
  CLKP ビット, 2-15  
  FSP ビット, 2-14  
  HALTR ビット, 2-74, 2-76

## C

CLK(R/X), 定義, A-1

CLKG, 定義, A-1  
CLKGDV, 定義, A-1  
CLKGDV(サンプル・レート生成器のデータ・ビッ  
ト・クロック・レート), 2-62  
CLKR, 定義, A-1  
CLKRM  
  受信クロックの選択, 2-65  
  定義, A-1  
CLKRP, 定義, A-1  
CLKS, 定義, A-2  
CLKS\_STAT, 定義, A-2  
CLKSM, 定義, A-2  
CLKSM(入力クロック・ソース・モード), 2-62  
CLKSP, 定義, A-2  
CLKSP(ビット・クロック極性), 2-62  
CLKSRG, 定義, A-2  
CLKSTP, 2-90  
  定義, A-2  
CLKSTP ビット/フィールド, 2-88  
CLKX, 定義, A-2  
CLKXM, 2-90  
  送信クロックの選択, 2-65  
  定義, A-2  
CLKXP, 2-90  
  定義, A-2  
CPU, 割り込み, 2-27  
CTMOD, 定義, A-2

## D

DATDLY, 定義, A-2  
DBLW, 定義, A-2  
DE, 定義, A-2  
DIND, 定義, A-2  
DINM, 定義, A-2  
DIR, 定義, A-2  
DIV クロック・モード, ~に切り替えた場合の  
  HPI-8 転送, 4-24

Index

## 索引

DLB, 定義, A-3  
DLB(デジタル・ループバック・モード), 2-64  
  受信クロックの選択, 2-65  
  受信フレーム同期の選択, 2-67  
DMA チャンネル, 2-85  
DMA チャンネル優先順位およびイネーブル制御レジスタ(DMPREC), 3-7  
DMA転送モード制御レジスタ(DMMCRn), 3-17  
  ビット/フィールドの説明, 3-17  
DMA 同期イベント, 3-13  
DMA 同期イベントおよびフレーム・カウント・レジスタ(DMSFCn), 3-13  
  ビット/フィールドの説明, 3-13  
DMA ブロック転送の割り込み発生モード, 3-28  
DMA メモリー・マップ  
  'C5402, 3-34  
  'C5410, 3-35  
  'C5420, 3-37  
DMA レジスタ, 3-6  
  DMA チャンネル優先順位およびイネーブル制御(DMPREC), ビット/フィールドの説明, 3-8  
DMA(ダイレクト・メモリー・アクセス)コントローラ, 1-2  
  FIFO 転送をサポートする設定, 6-4  
  FIFO 動作のための構成, 6-5  
  エミュレーション制御, 3-11  
  概要, 3-2  
  拡張 HPI のアクセス, 3-42  
  多重割り込み制御, 3-9  
  チャンネル, 2-85  
  チャンネル 0 および 1, FIFO 送受信の設定, 6-4  
  チャンネル・イネーブル制御, 3-9  
  チャンネル・コンテキスト・レジスタ, 3-11  
    エレメント・カウント・レジスタ, 3-12  
    転送元アドレス・レジスタと転送先アドレス・レジスタ, 3-11  
    同期イベントおよびフレーム・カウント・レジスタ(DMSFCn), 3-13  
  チャンネル転送速度の例, 3-40  
  チャンネル優先順位制御, 3-11  
  定義, A-3  
  転送, ダブルワード・モード, 3-16  
  転送サイクル・タイム, 3-39  
  転送先プログラム・ページ・アドレス・レジスタ(DMDSTP), 3-33

転送のレイテンシー, 3-39  
転送元プログラム・ページ・アドレス・レジスタ(DMSRCP), 3-33  
同期イベント, 3-13  
  'C5402, 3-14  
  'C5410, 3-15  
  'C5420, 3-16  
動作および設定, 3-4  
  パワーダウン・モードでの動作, 3-43  
  フレーム・カウント, 3-17  
  プログラム例, 3-44  
  ブロック転送の割り込み発生モード, 3-28  
メモリー・マップ  
  'C5402, 3-34  
  'C5410, 3-35  
  'C5420, 3-37  
レジスタ, 3-6  
  チャンネル優先順位およびイネーブル制御レジスタ(DMPREC), 3-7  
  転送先プログラム・ページ・アドレス・レジスタ(DMDSTP), 3-33  
  転送モード制御レジスタ(DMMCRn), 3-17  
  転送元プログラム・ページ・アドレス・レジスタ(DMSRCP), 3-33  
  同期イベントおよびフレーム・カウント・レジスタ(DMSFCn), 3-13  
  レジスタのサブアドレッシング, 3-4  
    自動インクリメントあり, 3-5  
    自動インクリメントなし, 3-5  
DMCTR, 定義, A-3  
DMD, 定義, A-3  
DMDST, 定義, A-3  
DMDSTP, 定義, A-3  
DMDSTP(DMA転送先プログラム・ページ・アドレス・レジスタ), 3-33  
DMFRI0, 定義, A-3  
DMFRI1, 定義, A-3  
DMGCR, 定義, A-3  
DMGDA, 定義, A-3  
DMGFR, 定義, A-3  
DMGSA, 定義, A-3  
DMIDX0, 定義, A-3  
DMIDX1, 定義, A-3  
DMMCR, 定義, A-3  
DMMCRn(DMA転送モード制御レジスタ), 3-17  
  ビット/フィールドの説明, 3-17

DMPREC, 定義, A-3  
DMPREC(DMA チャンネル優先順位およびイネーブル制御レジスタ), 3-7  
ビット/フィールドの説明, 3-8  
DMS, 定義, A-3  
DMSA, 定義, A-3  
DMSDI, 定義, A-3  
DMSDN, 定義, A-3  
DMSFC, 定義, A-3  
DMSFCn(DMA 同期イベントおよびフレーム・カウンタ・レジスタ), 3-13  
ビット/フィールドの説明, 3-13  
DMSRC, 定義, A-3  
DMSRCP, 定義, A-3  
DMSRCP(DMA転送元プログラム・ページ・アドレス・レジスタ), 3-33  
DPRC, 定義, A-3  
DR, 定義, A-4  
DRR, 定義, A-4  
DSP, 技術資料, vii, ix, x, xi, xii, xiii  
DSPINT, 定義, A-4  
DSPINT と HINT の動作, 4-23  
DSYN, 定義, A-4  
DX, 定義, A-4  
DX イネーブラ, 2-8  
A-bis モード, 2-37  
ノーマル・モード, 2-37  
DX ピン, 遅延イネーブル/ディセーブル, 2-37  
DX\_STAT, 定義, A-4  
DXENA, 定義, A-4  
DXR, 定義, A-4  
DXR2, 2-5

## E

EMIF/HPI-16 モード, 6-8  
EnhXIO(拡張外部バス・インターフェイス), 7-1

## F

FETCH, 定義, A-4  
FIFO  
～転送をサポートする DMA 設定, 6-4

～動作のための DMA 構成, 6-5  
構成, 'VC5420, 6-3  
送受信の設定, DMA チャンネル 0 および 1, 6-4  
双方向～, 6-3  
定義, A-4  
FPER, 定義, A-4  
FPER(フレーム周期), 2-66  
FREE, 定義, A-4  
FREE ビット, 2-95  
FRLen(R/X), 定義, A-4  
FRST, 定義, A-4  
FSG, 定義, A-4  
FSGM  
送信フレーム同期信号の選択, 2-68  
定義, A-5  
FSR, 定義, A-5  
FSRM  
受信フレーム同期の選択, 2-67  
定義, A-5  
FSRP, 定義, A-5  
FSX, 定義, A-5  
FSXM  
送信フレーム同期信号の選択, 2-68  
定義, A-5  
FSXP, 定義, A-5  
FWID, 定義, A-5  
FWID(フレーム幅), 2-66

## G

GPIO  
コードの例, 4-36  
定義, A-11  
GPIO 機能, 使用法, 4-36  
GPIOCR(汎用 I/O ステータス・レジスタ), 4-30  
GPIOCR(汎用 I/O 制御レジスタ), ビット機能, 4-30  
GPIOSR(汎用 I/O ステータス・レジスタ), 4-32  
ビット機能, 4-32  
GRST, 定義, A-5  
GSYNC  
受信フレーム同期の選択, 2-67  
定義, A-5

## H

 $\overline{\text{HAS}}$ 

- ～を使用しない HPIC リード, 5-11
- ～を使用しないホスト・アクセス, 5-11
- ～を使用する HPIAライト, 5-10
- ～を使用するホスト・アクセス, 5-10

HCNTL0/1 モード, 5-8

HINT, 定義, A-5

HMODE, 定義, A-5

## HPI

- 参照⇒ ホスト・ポート・インターフェイス  
のストローブおよび選択ロジック, 5-3

HPI インクリメント・モード, 定義, A-5

HPI 制御レジスタ(HPIC), ビット説明, 4-12

HPI へのライト・アクセス(自動インクリメントあり), 4-22

HPI-16, 1-2

HPI-8, 1-2

HPI-16(拡張 16 ビット・ホスト・ポート・インターフェイス), 5-1

- DMA との相互作用, 5-19

- DSP クロック・モードの変更, 5-22

- EMIF モード, 6-8

- PLL から DIV へのクロック・モード変更時の  
～, 5-22

- 外部メモリー空間からのアクセス, 6-7

- 外部メモリー・インターフェイスの接続,  
'VC5420, 6-7

## 動作

- IDLEn 時, 5-21

- リセット時, 5-21

- 動作の概要, 5-2

- ピンの説明, 5-4

- 分離モードでのインターフェイス, 'VC5420,  
5-15

- メモリー・マップ, 5-18

- HPI に対する～, 'VC5420, 5-18

HPI-8 アクセスのレイテンシー, 4-16

HPI-8(拡張 8 ビット・ホスト・ポート・インターフェイス), 4-1  
アクセス

- IDLE1 および IDLE2 モード, 4-26

- IDLE3 モード, 4-26

- アクセスシーケンスの例, 4-19, 4-28

- アクセス種類別の内部遅延, 4-17

- アクセスのレイテンシー, 4-16

アドレス・レジスタとメモリー・マップ, 4-9

ウェートステートの発生条件, 4-18

概要, 4-2

各信号の名称とその機能, 4-6

拡張アドレッシング, 4-10

基本機能について, 4-4

クロック・モード変更時の転送に関する考慮事  
項, 4-24

ストローブと選択ロジック, 4-8

制御レジスタ・ビットとその機能, 4-12

タイミング図, 4-15

データ・ピン, 汎用入出力ピンとしての, 4-30  
転送

- 'C54x が IDLE3 モードの場合, 4-27

- DIV クロック・モードに切り替えた場合,  
4-24

- クロック・モード変更時の考慮事項, 4-24

## 動作

- リセットが与える影響, 4-28

- リセット時('C5410), 4-29

動作について, 4-6

入力制御信号とその機能, 4-9

ホストからのリード/ライト・アクセス, 4-14

メモリー・マップ, 'VC5402 と 'VC5410, 4-9

ライト・アクセス(自動インクリメントあり),  
4-22

リセット後のアクセス, 4-28

リセット時のアクセス, 'C5410, 4-28

リード・アクセス(自動インクリメントあり),  
4-21

レジスタの説明, 4-5

HPIA, 定義, A-5

HPIA ライト,  $\overline{\text{HAS}}$  を使用する～, 5-10

## HPIC

定義, A-5

ビットの説明, 5-8

レジスタ, 5-8

HPIC のリード/ライト, ホストと 'C54x による,  
特徴, 4-13

HPIC リード,  $\overline{\text{HAS}}$  を使用しない, 5-11

HPIC レジスタ, ホストと 'C54x によるアクセ  
ス, 4-13

HPIC(HPI 制御レジスタ), ビットの説明, 4-12

HPID, 定義, A-5

HPIDライト, 自動インクリメントを使用, 5-13

HPID ライト, 分離モード, 5-17

HPIDリード, 自動インクリメントを使用, 5-12

HPID リード, 分離モード, 5-16

HPIENA, 定義, A-5

HRDY, 定義, A-6

HSTRB, 定義, A-6

## I

IDLE, 2-98

IDLE 使用時の考慮事項, 4-26

IDLE1 および IDLE2 モードでの HPI-8 アクセス, 4-26

IDLE3 モードでの HPI-8 アクセス, 4-26

IDLEn 時の HPI-16 の動作, 5-21

IMOD, 定義, A-6

INTOSEL, 定義, A-6

IPINT, 定義, A-6

IPIRQ, 定義, A-6

## L

LSB, 定義, A-6

## M

McBSP, 2-1

～を使用するサブシステム通信, 6-10

CPU 割り込み, 2-6

DMA イベント同期, 2-6

McBSP 間通信, 6-10

McBSP 動作

SPI スレーブとして, 2-92

SPI マスタとして, 2-90

パワーダウン・モード, 2-98

MCR1(マルチチャネル制御レジスタ 1), ビット/フィールドの説明, 2-73

SPI モードでの初期化, 2-93

インターフェイス信号, 2-4

概要, 2-3

基本的な動作, 2-38

クロック設定, 2-95

最大フレーム周波数, 2-39

受信/送信フレーム長の設定, 2-31

受信/送信ワード長の設定, 2-32

受信動作, 2-38

送信動作, 2-39

定義, A-12

特長, 2-2

内部ブロック図, 2-3

ピンの設定, 汎用 I/O, 2-97

ピンのリセット状態, 2-23

プログラミングのサンプル・プログラム, 2-99

マルチチャネル制御レジスタ 2(MCR2), 2-75

ビット/フィールドの説明, 2-75

リセット, 2-23

レジスタ, 2-5

MCR1(マルチチャネル制御レジスタ 1), 2-73

McBSP 間通信, 6-10

McBSP クロック設定, 2-95

McBSP 初期化, SPI モード, 2-93

McBSP 動作

SPI スレーブとして, 2-92

SPI マスタとして, 2-90

McBSP の CPU 割り込みおよび DMA イベント同期, 表, 2-6

McBSP の動作, パワーダウン・モード, 2-98

McBSP のレジスタ, 2-5

McBSP ピン, 汎用 I/O, 2-96

McBSP ピンの設定, 汎用 I/O, 表, 2-97

McBSP プログラミングのサンプル・プログラム, 2-99

McBSP リセット, 2-23

McBSPs, 1-2

MCR1(マルチチャネル制御レジスタ 1), 2-73  
ビット/フィールドの説明, 2-73

MCR2(マルチチャネル制御レジスタ 2), 2-75  
ビット/フィールドの説明, 2-75

MISO(マスタ・イン・スレーブ・アウト), 2-86

Mitel ST バス, 2-69

MOSI(マスタ・アウト・スレーブ・イン), 2-86

MP/MC ビット・レベル, リセット時, 6-8

MSB, 定義, A-6

## P

PCR, 定義, A-11

PCR(ピン制御レジスタ), 2-12

PLL から DIV へのクロック・モード変更時の  
HPI-16 の動作, 5-22

## R

RBR, 2-5  
定義, A-6  
RCBLK, 定義, A-6  
RCER, 定義, A-6  
RCERA (受信チャネル・イネーブル・レジスタ・  
パーティション A), 2-80  
ビット/フィールドの説明, 2-80  
RCERB (受信チャネル・イネーブル・レジスタ・  
パーティション B), 2-81  
ビット/フィールドの説明, 2-81  
RCOMPAND, 定義, A-6  
RCR, 定義, A-10  
RCR1(受信制御レジスタ 1), 2-16  
ビット/フィールドの説明, 2-16  
RCR2(受信制御レジスタ 2), 2-17  
ビット/フィールドの説明, 2-17  
RDATDLY, 2-34  
定義, A-6  
 $\overline{\text{RESET}}$ , 2-22  
定義, A-6  
REVT, 2-26  
定義, A-7  
REVTA, 定義, A-7  
REVTA(受信 A-bis イベント), 2-84  
RFIG, 定義, A-7  
RFRLEN(1,2), 2-31  
RFRLEN1, 定義, A-7  
RFRLEN2, 定義, A-7  
RFULL, 2-44  
定義, A-7  
RINT, 2-26  
定義, A-7  
RIOEN, 2-12  
定義, A-7  
RJUST, 定義, A-7  
RMCM, 定義, A-7  
RPABLK, 定義, A-7  
RPBBLK, 定義, A-7  
RPHASE, 定義, A-7  
RRDY, 2-26  
定義, A-7  
 $\overline{\text{RRST}}$ , 定義, A-7

$\overline{\text{RRST}}$  ビット, 2-93  
RSR, 定義, A-7  
rsvd, 定義, A-7  
RSYNCERR, 2-46  
定義, A-7  
RWDLEN(1,2), 2-31  
RWDLEN1, 2-90  
定義, A-7  
RWDLEN2, 定義, A-8

## S

SCK(シフト・クロック), 2-86  
SIND, 定義, A-8  
SOFT, 定義, A-8  
SOFT ビット, 2-10, 2-95  
SPCR, 定義, A-8  
SPCR1, ビット/フィールドの説明, 2-7  
SPCR2, ビット/フィールドの説明, 2-10  
SPI 構成  
スレーブとしての McBSP, 2-87  
マスタとしての McBSP, 2-87  
SPI スレーブとしての McBSP 動作, 2-92  
SPI 転送, CLKSTP = 10b および CLKXP = 0,  
2-88  
SPI 転送, CLKSTP = 10b および CLKXP = 1,  
2-89  
SPI 転送, CLKSTP = 11b および CLKXP = 0,  
2-89  
SPI 転送, CLKSTP = 11b および CLKXP = 1,  
2-89  
SPI プロトコル, McBSP クロック・ストップ・  
モード, 2-86  
SPI マスタとしての McBSP 動作, 2-90  
SPI モード, 2-8  
SPI モードでの McBSP 初期化, 2-93  
SPSA, 定義, A-8  
SRGR, 定義, A-8  
SRGR1(サンプル・レート生成器レジスタ 1),  
2-59  
ビット/フィールドの説明, 2-59  
SRGR2(サンプル・レート生成器レジスタ 2),  
2-60  
ビット/フィールドの説明, 2-60

SS(スレーブ・イネーブル), 2-86

ST バス・クロック

シングルレート, 2-70

ダブルレート, 2-69

ST バスと MVIP の例, 2-69

## T

TDM シリアル・ポート制御レジスタ(TSPC)

MCM ビット, 2-60

RRDY ビット, 2-9

$\overline{\text{RRST}}$  ビット, 2-9

TXM ビット, 2-13, 2-16, 2-17, 2-18,  
2-19, 2-20, 2-21, 2-61

XRDY ビット, 2-11

$\overline{\text{XRST}}$  ビット, 2-11

TDM データ・ストリーム, 定義, A-8

## W

WDLEN(R/X), 定義, A-8

## X

XCBLK, 定義, A-8

XCER, 定義, A-8

XCERA (送信チャネル・イネーブル・レジスタ・  
パーティション A), 2-81  
ビット/フィールドの説明, 2-81

XCERB (送信チャネル・イネーブル・レジスタ・  
パーティション B), 2-82  
ビット/フィールドの説明, 2-82

XCOMPAND, 定義, A-8

XCR1(送信制御レジスタ 1), 2-19  
ビット/フィールドの説明, 2-19

XCR2(送信制御レジスタ 2), 2-20  
ビット/フィールドの説明, 2-20

XDATDLY, 2-34  
定義, A-8

$\overline{\text{XEMPTY}}$ , A-8

$\overline{\text{XEMPTY}}$  ビット, 2-11

XEVT, 2-26  
定義, A-8

XEVTA, 定義, A-9

XEVTA イベント, 2-85

XEVTA(送信 A-bis イベント), 2-84

XFIG, 定義, A-9

XFRLLEN(1,2), 2-31

XFRLLEN1, 定義, A-9

XFRLLEN2, 定義, A-9

XHPIA, 定義, A-9

XINT, 2-26  
定義, A-9

XINTM, 定義, A-9

XIO2(外部パラレル・インターフェイス)

概要, 7-2

バス・タイミング, 7-3

不連続メモリー・リードおよび I/O リード,  
7-3

メモリー・ライトおよび I/O ライト, 7-5

連続メモリー・リード, 7-4

XIOEN, 2-12  
定義, A-9

XMCM, 定義, A-9

XMCM の動作, 図, 2-79

XPABLK, 定義, A-9

XPBBLK, 定義, A-9

XPHASE, 定義, A-9

XRDY, 2-26  
定義, A-9

XRDY ビット, 2-11

$\overline{\text{XRST}}$ , 定義, A-9

$\overline{\text{XRST}}$  ビット, 2-11, 2-93

XSR, 2-5  
定義, A-9

XSREMPY ビット, 2-11

XSYNCERR, 2-50  
定義, A-9

XSYNCERR ビット, 2-11

XWDLEN(1,2), 2-31

XWDLEN1, 2-90  
定義, A-9

XWDLEN2, 定義, A-9

## あ

アクセス, リセット後の HPI-8 への, 4-28

アクセス, リセット時の HPI-8 への, 4-28

アクセス・シーケンスの例(HPI-8), 4-19

アクセス・モード, 定義, A-10

圧伸

定義, A-10

内部データ, 2-55

フロー, 2-54

圧伸ハードウェアの動作: (R/X)COMPAND,  
2-53

アドレスの自動インクリメント, 4-11

アドレッシング・モード, 3-21

マルチフレーム・モード, 3-21

アプリケーション

医療, vii, xiii

開発サポート, vii, xiii

自動車, vii, xii

スピーチ/音声, vii, x

汎用 DSP, vii

民生機器, vii, xiii

## い

一般的な SPI インターフェイス, 2-86

イベント, XEVTA および REVTA, 2-84

医療アプリケーション, vii, xiii

## う

ウェートステートの発生条件, 4-18

ウェートステート発生器, 定義, A-11

## え

エミュレーション FREE ビットおよび SOFT  
ビット, 2-95

エミュレーション・ビット, 2-95

エミュレーション制御, DMA 機能に対する~,  
3-11

エレメント・カウント・レジスタ, 3-12

## お

オーバーランをとまなう受信, RFULL, 2-44

折り返しアドレスの算出法

インデックス・アドレッシングでのシングル  
ワード転送, 3-26

インデックス・アドレッシングでのダブルワー  
ド転送, 3-26

## か

開発サポート・アプリケーション, vii, xiii

外部のデータ通信, 2-4

外部のマスタ・クロック, 2-92

外部パラレル・インターフェイス(XIO2)

概要, 7-2

バス・タイミング, 7-3

不連続メモリー・リードおよび I/O リード,  
7-3

メモリー・ライトおよび I/O ライト, 7-5

連続メモリー・リード, 7-4

外部メモリー・インターフェイス, HPI-16 との接  
続, 'VC5420, 6-7

拡張 16 ビット・ホスト・ポート・インターフェイス  
(HPI-16), 5-1

拡張 8 ビット・ホスト・ポート・インターフェイス  
(HPI-8), 基本機能について, 4-4

拡張 8 ビット・ホスト・ポート・インターフェイス  
(HPI-8), 4-1

概要, 4-2

拡張アドレッシング, DMA, 3-33

拡張アドレッシング, HPI-8, 4-10

拡張外部バス・インターフェイス(EnhXIO), 7-1

拡張型プログラマブル・ウェートステート発生器,  
定義, A-10

画像処理アプリケーション, vii, ix

## き

休止状態, 2-98

共用モード, 5-7

HPI-16 へのインターフェイス, 'VC5420,  
5-7

定義, A-10

## く

クロック・ストップ・モード, 2-8

クロック・ストップ・モードの設定, 2-88

クロック・タイミングの例, 2-69

クロック・モードの変更による HPI-16 への影響,  
5-22



クロックおよびフレーム生成, 2-57  
 クロックとフレームの生成, サンプル・レート生成器, 2-58  
 軍需アプリケーション, vii, xi

## こ

効率的なデータ転送, 2-32  
 最大パケット周波数での~, 2-42  
 フレーム同期無視ビットによる~, 2-41

## さ

最大フレーム周波数, 2-39  
 受信/送信, 2-40  
 サブシステム, 定義, A-10  
 サブシステム通信, McBSP を使用する~, 6-10  
 サンプル・レート生成器, 2-92  
   クロックとフレームの生成, 2-58  
   図, 2-58  
   定義, A-10  
   データ・ビット・クロック・レート(CCLKGDV), 2-62  
   リセット手順, 2-61  
 サンプル・レート生成器リセット・ビット(GRST), 2-93  
 サンプル・レート生成器レジスタ 1(SRGR1), 2-59  
   ビット/フィールドの説明, 2-59  
 サンプル・レート生成器レジスタ 2(SRGR2), 2-60  
   ビット/フィールドの説明, 2-60

## し

自動インクリメント  
   ~を使用する HPID ライト, 5-13  
   ~を使用する HPID リード, 5-12  
 自動インクリメント動作, 5-12  
 自動車アプリケーション, vii, xii  
 自動初期化, 3-27  
 自動バッファリング(ABU)モード, 3-23  
 自動バッファリング設定, 2-98  
 シフト・クロック(SCK), 2-86

受信/送信ワード長の設定, 2-32  
 受信クロック信号(BCLKR), 2-86  
 受信クロックの選択: DLB, CLKRM, 2-65  
 受信シフト・レジスタ(RSR), 2-4  
 受信制御レジスタ 1(RCR1), 2-16  
   ビット/フィールドの説明, 2-16  
 受信制御レジスタ 2(RCR2), 2-17  
   ビット/フィールドの説明, 2-17  
 受信チャンネル・イネーブル・レジスタ・パーティション A(RCERA), 2-80  
   ビット/フィールドの説明, 2-80  
 受信チャンネル・イネーブル・レジスタ・パーティション B(RCERB), 2-81  
   ビット/フィールドの説明, 2-81  
 受信データとクロック, 2-30  
 受信データの位置調整と符号拡張, RJUST, 2-52  
 受信動作, 2-22  
 受信バッファ・レジスタ(RBR), 2-4  
   定義, A-10  
 受信部リセット, McBSP 動作, 2-93  
 受信フレーム同期信号(BFSR), 2-86  
 受信フレーム同期の選択: DLB, FSRM, GSYNC, 2-67  
 受信フレーム同期パルスへの応答, 2-47  
 受信レディ状態, 2-26  
 情報通信アプリケーション, vii, xii  
 シリアル・データ入力(マスタ・イン・スレーブ・アウト, MISO), 2-86  
 シリアル・データ出力(マスタ・アウト・スレーブ・イン, MOSI), 2-86  
 シリアル・ポート, 1-2  
   受信オーバーラン, 2-45  
   受信オーバーランの回避, 2-46  
   設定, 2-6  
   リセット, 2-22  
   例外条件, 2-43  
 シリアル・ポート・インターフェイス, 2-2  
 シリアル・ポート制御レジスタ, 2-6  
 シリアル・ポート制御レジスタ 1(SPCR1)  
   図, 2-7  
   ビット/フィールドの説明, 2-7  
 シリアル・ポート制御レジスタ 1(SPCR1), 2-88  
 シリアル・ポート制御レジスタ 2(SPCR2), 2-93  
   ビット/フィールドの説明, 2-10

## 索引

シリアル・ポート制御レジスタ(SPC)  
DLB ビット, 2-7  
RSRFULL ビット, 2-9  
シリアル・ポート制御レジスタ(SPCR), 定義,  
A-10  
シングル・フェーズ・フレーム  
1つの32ビット・ワード構成, 2-34  
4つの8ビット・ワード構成, 2-33  
シングルレート ST バス・クロック, 2-70  
シングルレート・クロックの例, 2-70

## す

スレーブ, 定義, A-10  
スレーブ・イネーブル信号(SS), 2-86

## せ

制御アプリケーション, vii, x  
設定, シリアル・ポート, 2-6

## そ

送受信制御レジスタ(RCR および XCR), 定義,  
A-10  
送信, データ上書きをとまなう~, 2-48  
送信エンプティ  
XEMPTY, 2-48  
回避, 2-50  
図, 2-49  
送信クロック信号(BCLKX), 2-86  
送信クロックの選択: CLKXM, 2-65  
送信シフト・レジスタ(XSR), 2-4  
送信制御レジスタ 1(XCR1), 2-19  
ビット/フィールドの説明, 2-19  
送信制御レジスタ 2(XCR2), 2-20  
ビット/フィールドの説明, 2-20  
送信チャネル・イネーブル・レジスタ・パーティショ  
ン A(XCERA), 2-81  
ビット/フィールドの説明, 2-81  
送信チャネル・イネーブル・レジスタ・パーティショ  
ン B(XCERB), 2-82  
ビット/フィールドの説明, 2-82  
送信動作, 2-22

送信部リセット, McBSP 動作, 2-93  
送信部リセット・ビット(XRST), 2-93  
送信フレーム同期信号の選択: FSXM, FSGM,  
2-68  
送信フレーム同期への応答, 2-51  
送信レディ状態, 2-26  
双方向 FIFO, 6-3

## た

ダイレクト・メモリー・アクセス(DMA)コントロー  
ラ, 1-2  
定義, A-11  
多重割り込み制御, DMA, 3-9  
多重割り込みの割り当て  
'C5402, 3-10  
'C5410, 3-10  
'C5420, 3-10  
ダブルレート ST バス・クロック, 2-69  
ダブルレート・クロックの例, 2-71  
ダブルワード・モード, DMA 転送, 3-16

## ち

チャネル・イネーブル・レジスタ:  
(R/X)CER(A/B), 2-80  
チャネル・イネーブル制御, DMA, 3-9  
チャネル・コンテキスト・レジスタ, DMA, 3-11  
チャネル選択の変更, 2-82  
チャネルのマスクとイネーブル, 2-76

## て

デジタル・ループバック・モード(DLB), 2-64  
データ・クロックの生成, 2-62  
データ受信(DR)ピン, 2-4  
データ受信レジスタ(DRR), 2-4  
データ送受信フロー, 2-22  
データ送信(DX)ピン, 2-4  
データ遅延, 2-34  
フレーミング・ビット破棄に使用される2ビッ  
トの~, 2-35  
データのソート  
ABU モード, 3-23

アドレス修飾による, 3-22

例, 3-23

デバイス・クロック, 2-98

デバイス・リセット, 2-23

McBSP の動作, 2-93

デュアル・フェーズ・フレームの例, 2-30

転送サイクル・タイム, DMA, 3-39

転送のレイテンシー, DMA, 3-39

転送元アドレス・レジスタと転送先アドレス・レジスタ, DMA, 3-11

転送元アドレスと転送先アドレスの選択と修飾, 3-20

## な

内部でのプロセッサ間 FIFO による通信, '5420, 3-43

内部でのプロセッサ間通信, 6-1

マルチコア DSP 内, 6-2

内部でのプロセッサ間割り込み, 6-11

内部のデータ通信, 2-4

## に

入力クロック・ソース・モード(CLKSM), 2-62

## は

パケット長, 2-90

はじめに, 1-1-1-8

バッファド・シリアル・ポート, 1-2

パーティション A および B におけるブロックによるチャンネルのイネーブル, 2-77

パワーダウン・モード, 2-98

汎用 DSP アプリケーション, vii

汎用 I/O, McBSP ピン, 2-96

汎用 I/O ステータス・レジスタ(GPIOSR), 4-32  
ビット機能, 4-32

汎用 I/O 制御レジスタ(GPIOCR), 4-30  
ビット機能, 4-30

汎用 I/O としてのピンの設定, 表, 2-97

汎用入力/出力ピン(GPIO), 定義, A-11

## ひ

ビット, FREE および SOFT, 2-95

ビット・オーダリング, 2-56

定義, A-11

ビット・クロック極性(CLKSP), 2-62

ビット・クロックとフレーム同期, 2-63

ピン制御レジスタ(PCR), 2-6, 2-12, 2-88, A-11

ビット/フィールドの説明, 2-12

## ふ

フリー・ラン, 2-95

フリー・ランニング・モード, 2-10

フレーム・カウント

DMA, 3-17

定義, A-11

フレーム・フォーマット, デュアル・フェーズ:

AC97, 2-36

フレームおよびクロック

設定, 2-27

動作, 2-28

フレーム検出, 初期化のための~, 2-68

フレーム周期(FPER), 2-66

フレーム長, 2-31

フレーム同期信号の生成, 2-4, 2-66

フレーム同期生成器リセット, 2-10

フレーム同期の無視, 2-40

~と予定外のフレーム同期パルス, 2-42

フレーム同期フェーズ, 2-30

フレームの例, 複数フェーズ: AC97, 2-35

フレーム幅(FWID), 2-66

プログラマブル・ウェイトステート発生器, 定義, A-11

プログラマブル・クロックおよびフレーム生成, 2-57

プログラマブル・バンクスイッチング・モジュール, 定義, A-11

プログラミングのサンプル・プログラム, 2-99

プログラム可能なフレーム周期とフレーム幅, 2-67

ブロック図, McBSP, 2-3

分離モード, 5-15

HPID ライト, 5-17

HPID リード, 5-16  
定義, A-11

## へ

ペリフェラル

ホスト・ポート・インターフェイス, 1-2  
ホスト・ポート・インターフェイス(HPI), 4-2

## ほ

ホスト, 定義, A-11  
ホスト・アクセス, HAS を使用しない~, 5-11  
ホスト・アクセス, HAS を使用する~, 5-10  
ホスト・デバイスによる 'C54x への割り込み,  
DSPINT を使用した, 4-23  
ホスト・ポート・インターフェイス, 1-2, 4-2  
一般的なシステム・ブロック図, 4-4  
ブロック図, 4-3  
ホスト・ポート・インターフェイス(HPI-16),  
A-11  
ホスト・ポート・インターフェイス(HPI-8), 定義,  
A-11  
ホストから HPI-8 へのリード/ライト・アクセス,  
4-14  
ホストと 'C54x によるアクセス, HPIC レジス  
タ, 4-13

## ま

マスタ, 定義, A-12  
マスタスレーブ構成, 2-86  
マルチコア DSP 内での通信, 6-2  
マルチサブシステム DSP, 6-2  
マルチチャネル・バッファド・シリアル・ポート,  
1-2  
マルチチャネル・バッファド・シリアル・ポート  
(McBSP), 2-1  
定義, A-12  
特長, 2-2  
マルチチャネル制御レジスタ 1(MCR1), 2-73  
ビット/フィールドの説明, 2-73  
マルチチャネル制御レジスタ 2(MCR2), 2-75  
ビット/フィールドの説明, 2-75

マルチチャネル選択動作, 2-72  
マルチチャネル選択のイネーブル, 2-76  
マルチチャネル動作の制御レジスタ, 2-73  
マルチメディア・アプリケーション, vii, xi

## み

民生機器アプリケーション, vii, xiii

## め

メモリー・マップ  
HPI に対する~, 'VC5420, 5-18  
HPI-16, 5-18

## よ

予定外の受信同期パルス, 2-48  
予定外の受信フレーム同期, RSYNCERR, 2-46  
予定外の送信フレーム同期, XSYNCERR, 2-50  
予定外の送信フレーム同期パルス, 2-52

## り

リセット  
～時の HPI-16 の動作, 5-21  
～時の HPI-8 の動作, 'C5410, 4-29  
～時の  $\overline{\text{MP}}/\overline{\text{MC}}$  ビット・レベル, 6-8  
HPI-8 の動作に与える影響, 4-28  
リセット状態, McBSP ピン, 2-23  
リード/ライトの特徴, ホストと 'C54x による  
HPIC への, 4-13  
リード・アクセス(自動インクリメントあり),  
HPI-8, 4-21

## れ

レイテンシー, 定義, A-12  
レジスタ  
受信シフト・レジスタ(RSR), 2-4  
受信制御レジスタ 1, 2-16  
受信制御レジスタ 2, 2-17  
受信バッファ・レジスタ(RBR), 2-4  
シリアル・ポート制御レジスタ 1, 2-7  
シリアル・ポート制御レジスタ 2, 2-10

送信シフト・レジスタ(XSR), 2-4  
データ送信レジスタ(DXR), 2-4  
ピン制御レジスタ(PCR), 2-12  
レジスタ・ビット値, SPI スレーブ動作用, 2-92  
レジスタ・ビット値, SPI マスタ動作用, 2-91  
レジスタ・ビット値, SPI モード設定用, 2-90  
レジスタのサブアドレッシング, 3-4  
    サブアドレッシング使用例(自動インクリメント  
        あり), 3-5  
    サブアドレッシング使用例(自動インクリメント  
        なし), 3-5

レディ状態, McBSP 受信部, 2-26  
レディ状態の決定, 2-26

## わ

ワード長, 2-31, 2-32  
割り込み, 内部でのプロセッサ間, 6-11  
割り込みの更新, 2-83  
割り込み発生, DMA, 3-27

## 索引

---

## 日本テキサス・インスツルメンツ株式会社

本 社 〒160-8366 東京都新宿区西新宿6丁目24番1号 西新宿三井ビルディング3階

|         |                                                                                                    |                          |                   |                   |
|---------|----------------------------------------------------------------------------------------------------|--------------------------|-------------------|-------------------|
| 大阪営業所   | 〒530-6026                                                                                          | 大阪市北区天満橋1丁目8番30号         | OAPオフィスタワー26階     | ☎06(6356)4500(代表) |
| 名古屋営業所  | 〒460-0003                                                                                          | 名古屋市中区錦2丁目4番3号           | 錦パークビル10階         | ☎052(232)5601(代表) |
| 松本営業所   | 〒390-0815                                                                                          | 長野県松本市深志1丁目2番1号          | 松本昭和ビル6階          | ☎0263(33)1060(代表) |
| 立川営業所   | 〒190-0012                                                                                          | 東京都立川市曙町2丁目3番12号         | ファースト立川センタースクエア1階 | ☎0425(27)6760(代表) |
| 京都営業所   | 〒600-8216                                                                                          | 京都市下京区塩小路通西洞院東入東塩小路843-2 | 日本生命京都ヤサカビル105階   | ☎075(341)7713(代表) |
| 大宮営業所   | 〒330-0802                                                                                          | 埼玉県大宮市宮町1丁目3番1号          | 野村不動産大宮共同ビル4階     | ☎048(647)2622(代表) |
| 米国駐在事務所 | TEXAS INSTRUMENTS Inc., 8505 Forest Lane, Dallas, 75243, U.S.A (TI Japan Ltd., U.S.A Sales Office) |                          |                   | ☎1-972-480-7464   |

埼玉県・鳩ヶ谷市 大分県・日出町 茨城県・美浦村 静岡県・小山町(制御機器事業部)

株式会社テキサス・インスツルメンツ筑波開発センター 茨城県・つくば市

■お問い合わせ先

プロダクト・インフォメーション・センター(PIC) ☎ ☎ 0120-81-0026 FAX ☎ ☎ 0120-81-0036

E-mail: pic-japan@ti.com

# ***TMS320C54x DSP***

高性能ペリフェラル

リファレンス・セット Volume.5

第1版 1999年 12月

---

発行所 日本テキサス・インスツルメンツ株式会社

東京都新宿区西新宿6-24-1 (西新宿三井ビルディング)

〒160-8366



