Optimal Implementation of MPEG4 HEAAC v2 Decoder on C64x+ DSPs

Tejaswi N &
Ameya Potadar

Ittiam Systems Pvt. Ltd.
tejaswi.n@ittiam.com
ameya.potadar@ittiam.com

Agenda

- Overview of HEAAC v2 technology
- Generic optimization approach
- Optimization details of HEAAC v2 decoder blocks
- Conclusions

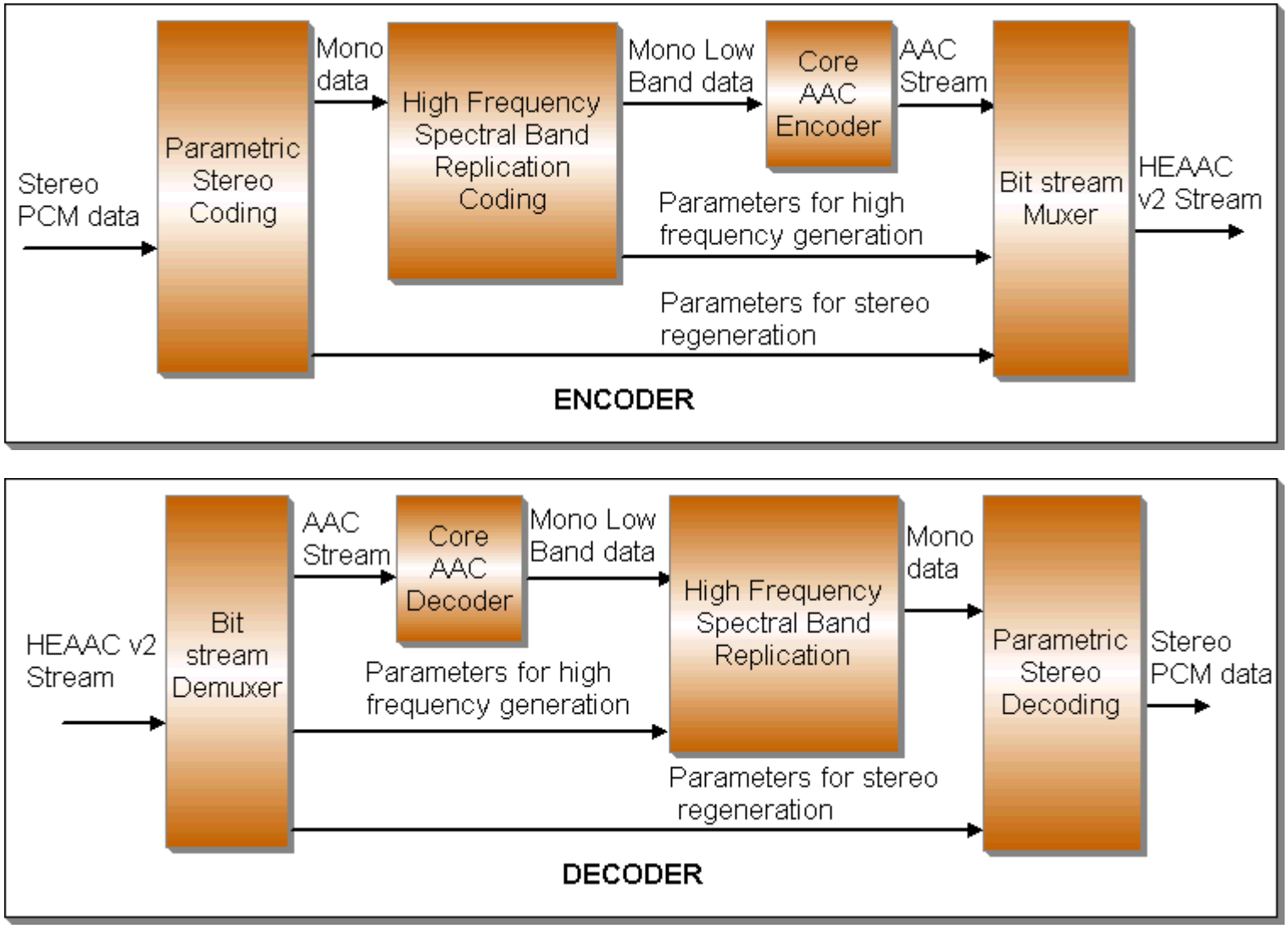
Agenda

- Overview of HEAAC v2 technology
- Generic optimization approach
- Optimization details of HEAAC v2 decoder blocks
- Conclusions

HEAAC v2

- AAC - a psycho acoustic based transform codec
- Transparent quality at 128 kbps, good quality at 96 kbps for stereo 44.1kHz audio streams
- Combination of SBR and AAC called HEAAC v1, here high frequency part of the spectrum is replicated from low-band
- Good quality at 48 to 64 kbps, used widely in DMB applications
- Combination of AAC, SBR and PS called HEAAC v2, here stereo channel redundancy is exploited
- Good quality at 32 kbps, used in mobile streaming

HEAAC v2


Stereo PCM data
Parametric Stereo Coding
Mono data
High Frequency Spectral Band Replication Coding
Mono Low Band data
Core AAC Encoder
AAC Stream
Bit stream Muxer
HEAAC v2 Stream
Parameters for high frequency generation
Parameters for stereo regeneration
ENCODER
HEAAC v2 Stream
Bit stream Demuxer
AAC Stream
Core AAC Decoder
Mono Low Band data
High Frequency Spectral Band Replication
Mono data
Parametric Stereo Decoding
Stereo PCM data
Parameters for high frequency generation
Parameters for stereo regeneration
DECODER

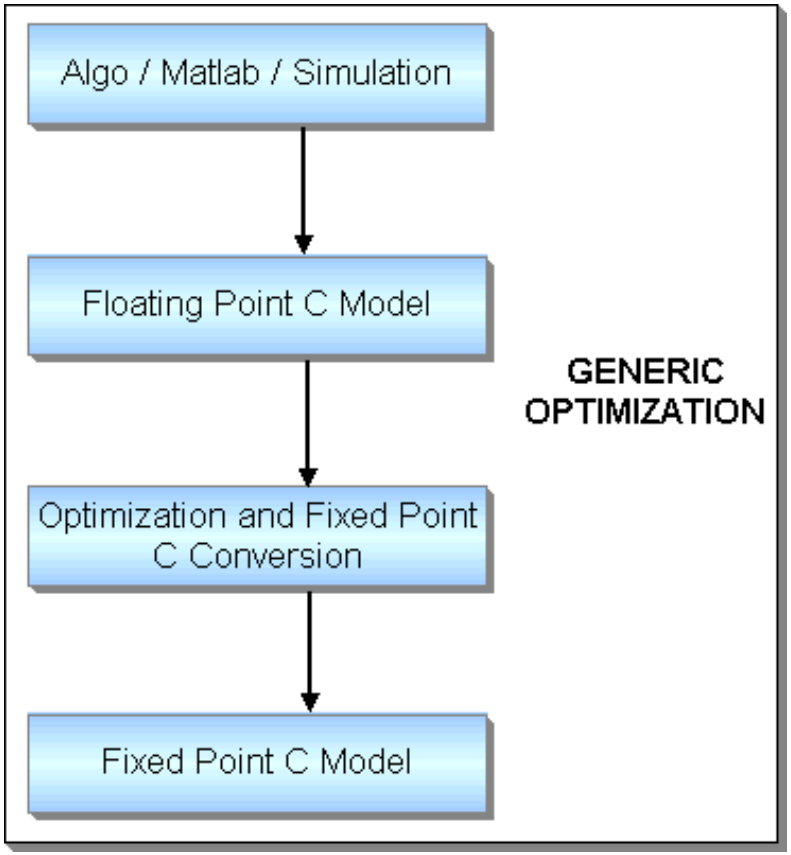
HEAAC v2

- **AAC Decoder**
  - Entropy decoding, Inverse quantize, Apply encoder tools, IMDCT
- **SBR Decoder**
  - Analysis to QMF (Quadrature Mirror Filter) bank domain, HF (High Frequency) generation, HF adjustment, HF addition, Synthesis to time domain
- **PS Decoder**
  - Analysis to higher resolution QMF banks, De-correlation, Apply rotation, Hybrid synthesis

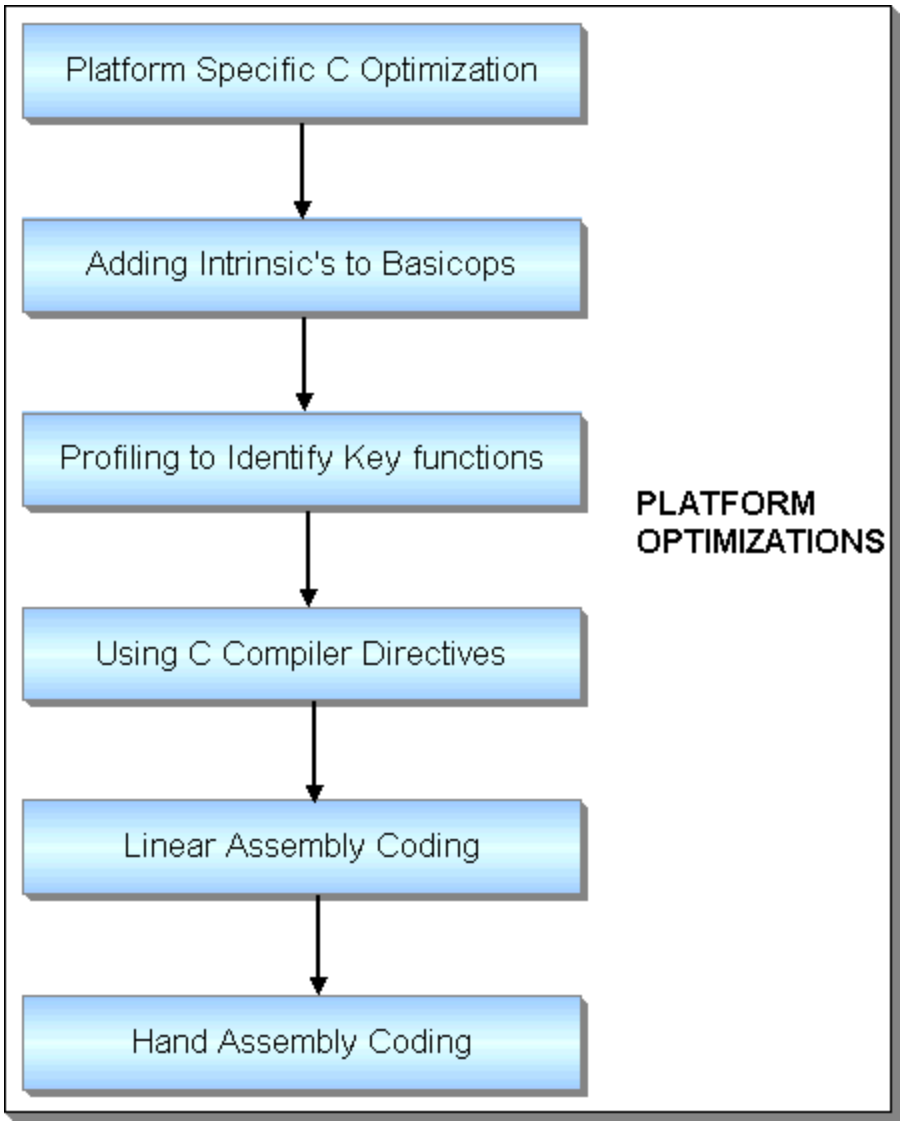
Agenda

- Overview of HEAAC v2 technology
- Generic optimization approach
- Optimization details of HEAAC v2 decoder blocks
- Conclusions

Codec Development



Codec Development



Optimization Phases

- **Compile on CCS the fixed point CMODEL**
  - Build options used
    - -O3
    - No Debug
    - -mt
    - Speed Most Critical
- **Add intrinsic to Basic Operations**
  - Commonly used intrinsic
    - _sshl, _smpy, _sshvl, _sshvr, _min2, _max2, _sadd, _ssub, _norm, _abs, _mpylir, _mpyhir, _mpyluhs

Optimization Phases

- **Profile breakup to find high complexity modules**
  - Classify modules to different types
    - Only C optimizations
    - Standard Library available (DSPLIB)
    - Linear assembly coding required
    - Hand assembly coding required

Optimization Phases

- **Standard Library usage for available functions**
  - E.g. FFT
- **C level optimizations**
  - Change C code to suit c64x instructions (shifts, mults)
  - Removing indirection from loops
  - “restrict” keyword usage
  - Specify double word alignment
  - #pragma MUST_ITERATE and UNROLL

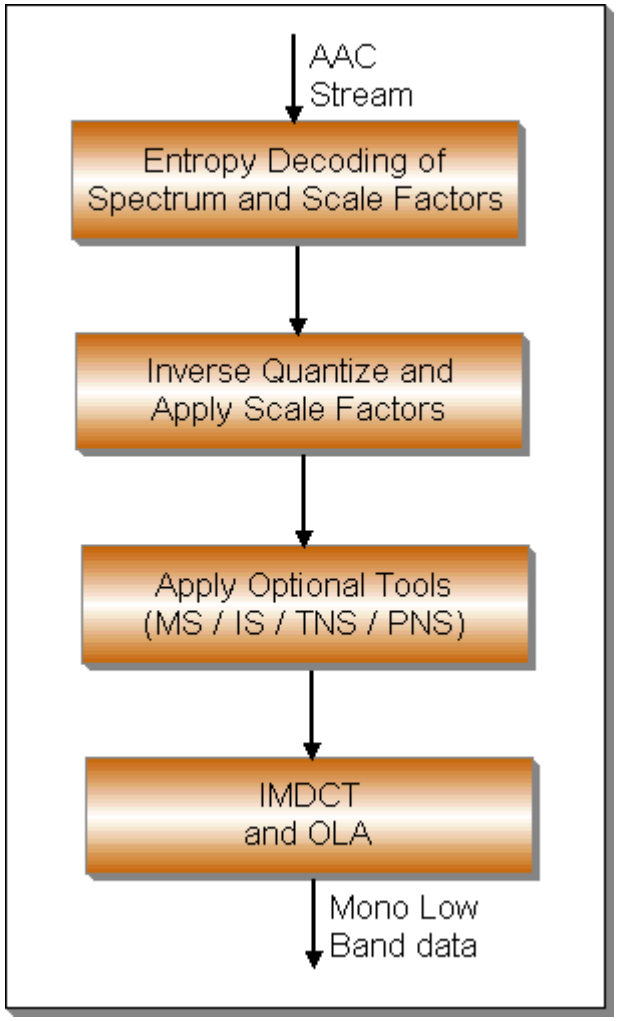
Optimization Phases

- **Linear assembly coding**
  - Coded if addition of intrinsic has not lead to best kernels or if SIMD instructions can be used
  - Compiler feed back used to reduce bottlenecks
- **Hand assembly coding**
  - Should be used as a last resort
  - Coded only if nested loops or control code present
  - Prologue and Epilogue merge of outer loop

Agenda

- Overview of HEAAC v2 technology
- Generic optimization approach
- Optimization details of HEAAC v2 decoder blocks
- Conclusions

AAC Decoder Blocks

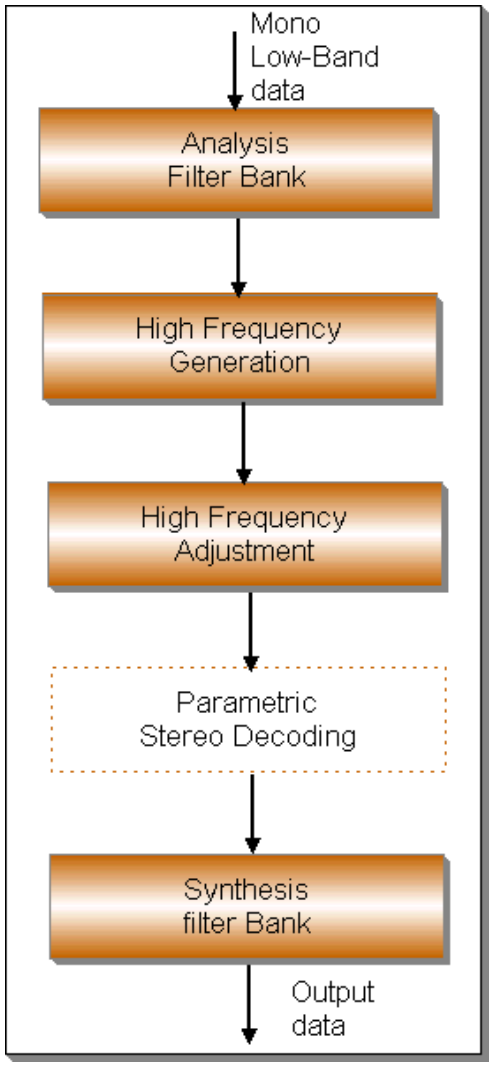


- **Entropy Decoding**
  - Huffman decoding, Loop with control code
- **Inverse quantization**
  - Power (x, 4/3), Loop with lookup table and mults
  - Apply scale factors x * power(2, sf/4), Loop with shifts and mults
- **Encoder optional tools**
  - IS/MS: stereo redundancy, Loop with add, sub and mults
  - TNS: better transient coding, Nested loop of IIR filtering
  - PNS: Noise generation, Loop with mults
- **IMDCT and OLA**
  - Transformation to time domain (PreTwiddle+FFT+PostTwiddle), Loops with mults, shifts and adds

AAC Decoder Blocks

Module	Description	Approach	Initial MCPS	Final MCPS	Reduction
Entropy Decoding of spectral and scale factor data	Control code	Intrinsic C and Compiler Directives	3.0	0.7	4.3x
Inverse Quantize and Apply Scale Factors	Loops with mac and shifts	Intrinsic C and Compiler Directives	1.5	0.4	3.8x
IS, MS, PNS	Loops with add, mults, norms	Intrinsic C and Compiler Directives	0.5	0.1	5.0x
TNS	Nested loop with mac and shifts	Hand Assembly Coded	1.5	0.4	3.8x
FFT	Loops with add, mults and shifts	Standard library	3.0	0.4	7.5x
Pre/Post twiddle for IMDCT and OLA	Loops with mac and shifts	Intrinsic C and Compiler Directives	1.0	0.3	3.3x
		Total	10.5	2.3	4.6x

SBR Decoder Blocks

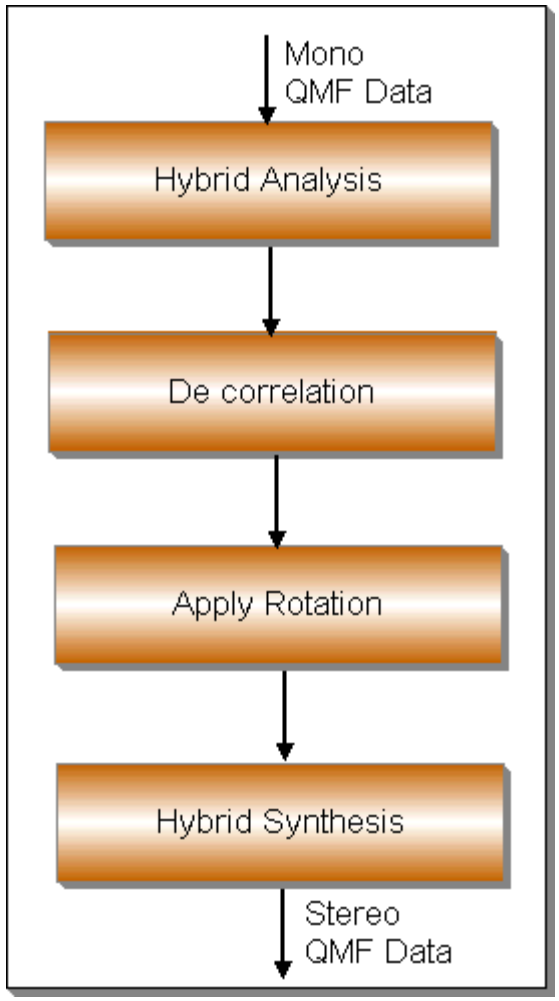


- **Analysis Filter bank**
  - Window and Add, Loop with 16x16 macs
  - DCT as pre/post twiddle and FFT, Loops with mults, shifts and adds
- **HF Generation**
  - Auto correlation and LPC calculation, Loop with macs and control code
  - LPC filtering, Loop with macs
- **HF adjustment**
  - Calculation of signal and noise/tone gains, control code
  - Adjust time slot, Loop with mult and adds
- **Synthesis Filter bank**
  - DCT as pre/post twiddle and FFT, Loops with mults, shifts and adds
  - Window and Add, Loop with 16x16 macs

SBR Decoder Blocks

Module	Description	Approach	Initial MCPS	Final MCPS	Reduction
Analysis and Synthesis Filter bank DCT twiddling and rescaling	Loops with add, mults and shifts	Intrinsic C and Compiler Directives	8.0	1.9	4.2x
Analysis and Synthesis window and add	Loops with 16x16 mac	Linear Assembly Coded	2.0	0.6	3.3x
High frequency generation	Loops with add, mults and shifts	Intrinsic C and Compiler Directives	3.0	0.5	6.0x
High frequency adjustment	Loops with add, mults, shifts and control code	Intrinsic C and Compiler Directives	7.0	2.1	3.3x
		Total	20	5.1	3.9x

PS Decoder Blocks



- **Hybrid Analysis**
  - Further analysis of lower bands for better frequency resolution, Loops with macs
- **De-correlation**
  - Generation of orthogonal vector
  - All pass filter with constant phase delays for each sub band , Loops with 16x16 macs
- **Apply Rotation**
  - Using the ICC and IID parameters to generate the rotation matrix, Loops with mults, table lookups and adds
  - Apply rotation matrix on the orthogonal vectors to get back the original intensity and time difference, Loops with mults, adds and shifts
- **Hybrid Synthesis**
  - Synthesis of lower bands, Loop with adds

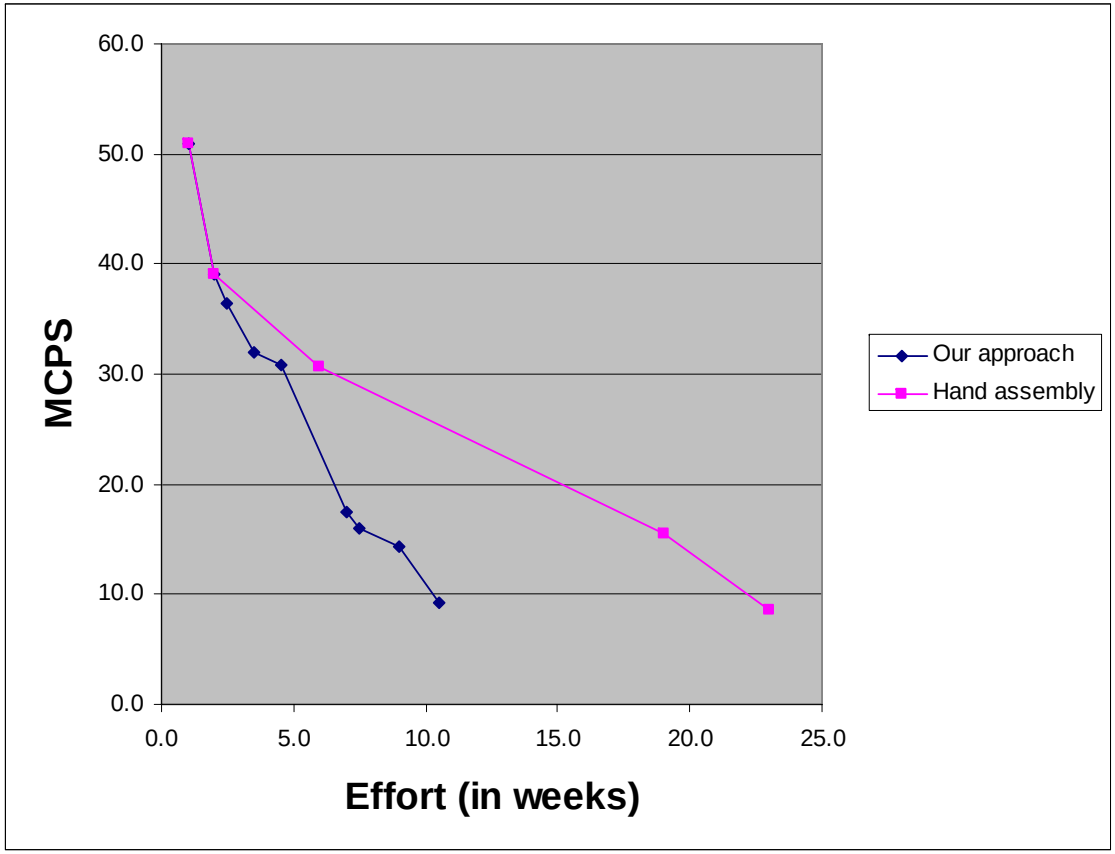
PS Decoder Blocks

Module	Description	Approach	Initial MCPS	Final MCPS	Reduction
Hybrid Analysis	Loops with add and mults	Intrinsic C and Compiler Directives	2.0	0.5	4.0x
De correlation	Loops with 16x16 mac	Linear Assembly Coded	5.3	1.0	5.3x
Apply Rotation	Loops with add, mults and shifts	Linear Assembly Coded	1.0	0.3	3.3x
Hybrid Synthesis	Loops with adds	Intrinsic C and Compiler Directives	0.3	0.1	3.0x
		Total	8.6	1.9	4.5x

Agenda

- Overview of HEAAC v2 technology
- Generic optimization approach
- Optimization details of HEAAC v2 decoder blocks
- Conclusions

MCPS vs. Effort

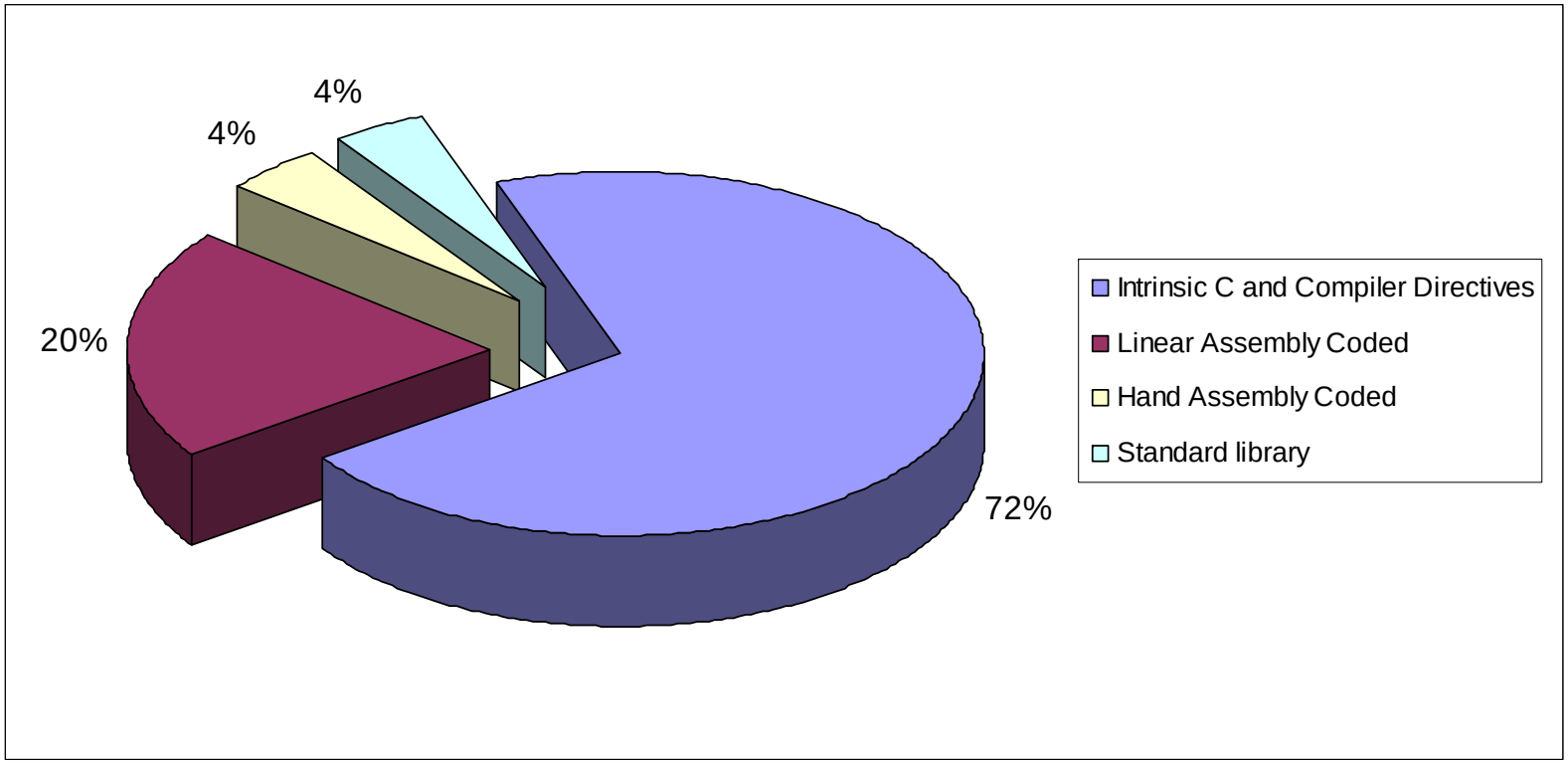


MCPS vs. Effort

Our approach		
Stages	**MCPS**	**Effort (in weeks)**
Cross compile on CCS	51.0	1.0
Intrinsic for Basicops	39.1	1.0
Standard Library	36.5	0.5
Compiler Directives - AAC part	32.0	1.0
Hand Asm - AAC part	30.9	1.0
Compiler Directives - SBR part	17.4	2.5
Linear Asm - SBR part	16.0	0.5
Compiler Directives - PS part	14.3	1.5
Linear Asm - PS part	9.3	1.5

Hand assembly optimization		
Stages	**MCPS**	**Effort (in weeks)**
Cross compile on CCS	51.0	1.0
Intrinsic for Basicops	39.1	1.0
Hand Asm - AAC Part	30.6	4.0
Hand Asm - SBR Part	15.5	13.0
Hand Asm - PS Part	8.6	4.0

MCPS Percentage Breakup



MCPS

Description of stream	MCPS (Peak)	MCPS (Average)
AAC only	5.0	4.1
AAC+SBR	8.9	6.1
AAC+SBR+PS	9.3	8.2

Conclusions

- **Best numbers achieved with C level optimizations itself**
  - Minimal effort (reduced time to market)
  - Better code maintenance
- **Hand assembly coding if significant reduction expected**

References

- ISO/IEC 14496-3:2001, Information technology - Coding of audio-visual objects - Part 3: Audio
- ISO/IEC 14496-3:2001/Amd.1:2003, Bandwidth Extension
- ISO/IEC 14496-3:2001/Amd.2:2004, Parametric Coding for High Quality Audio
- TMS320C6000 Optimizing Compiler User's Guide (spru187k)
- TMS320C64x/C64x+ DSP CPU and Instruction Set Reference Guide (spru732b)

TI Developer Conference

Optimal Implementation of MPEG4 HEAAC v2 decoder on C64x+ DSPs

Tejaswi N &
Ameya Potadar

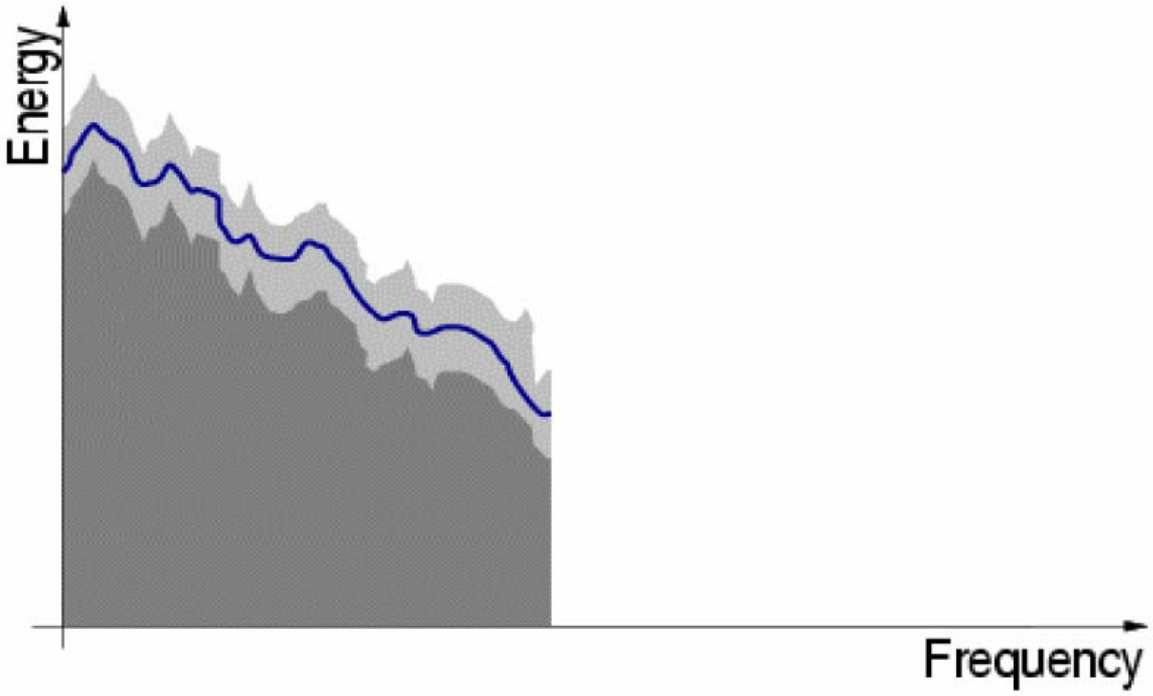
Ittiam Systems Pvt. Ltd.
tejaswi.n@ittiam.com
ameya.potadar@ittiam.com

SEE THE FUTURE

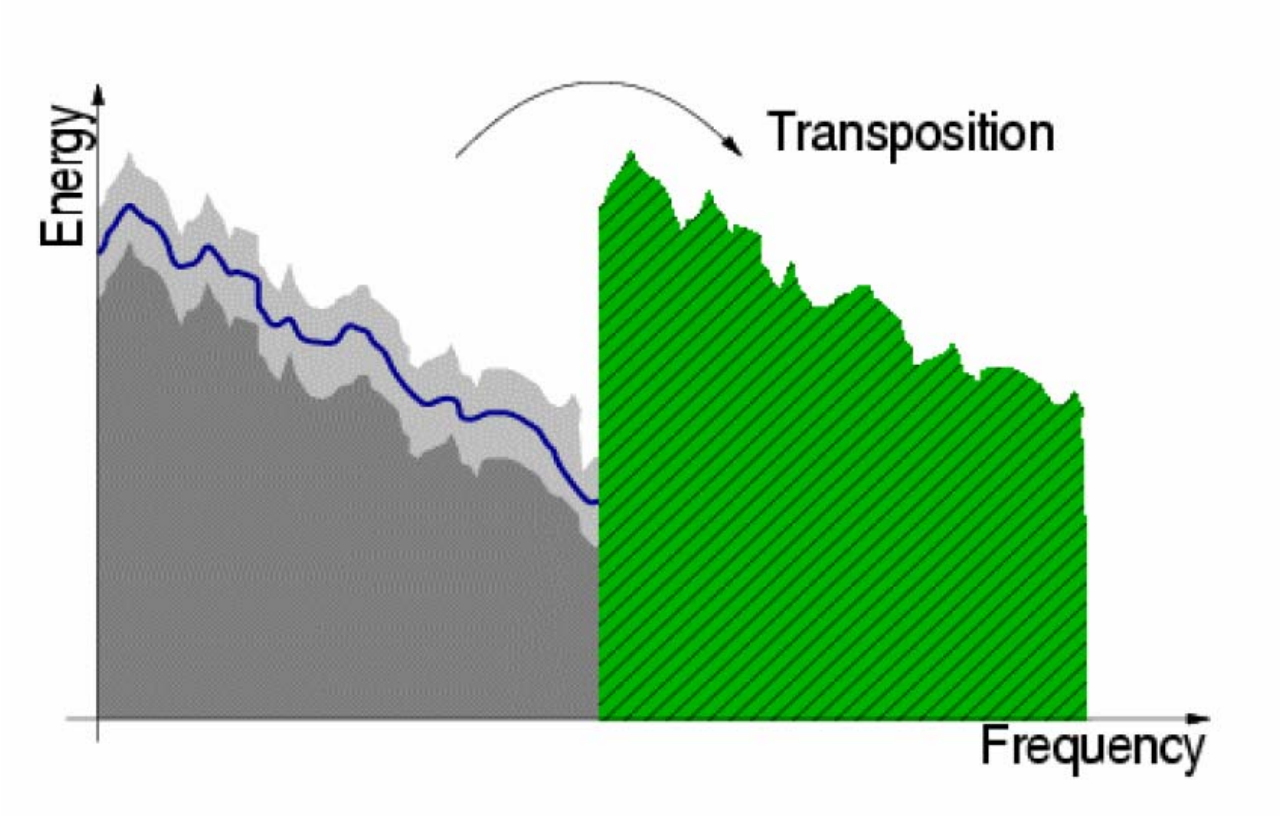
CREATE YOUR OWN

Backup Slides

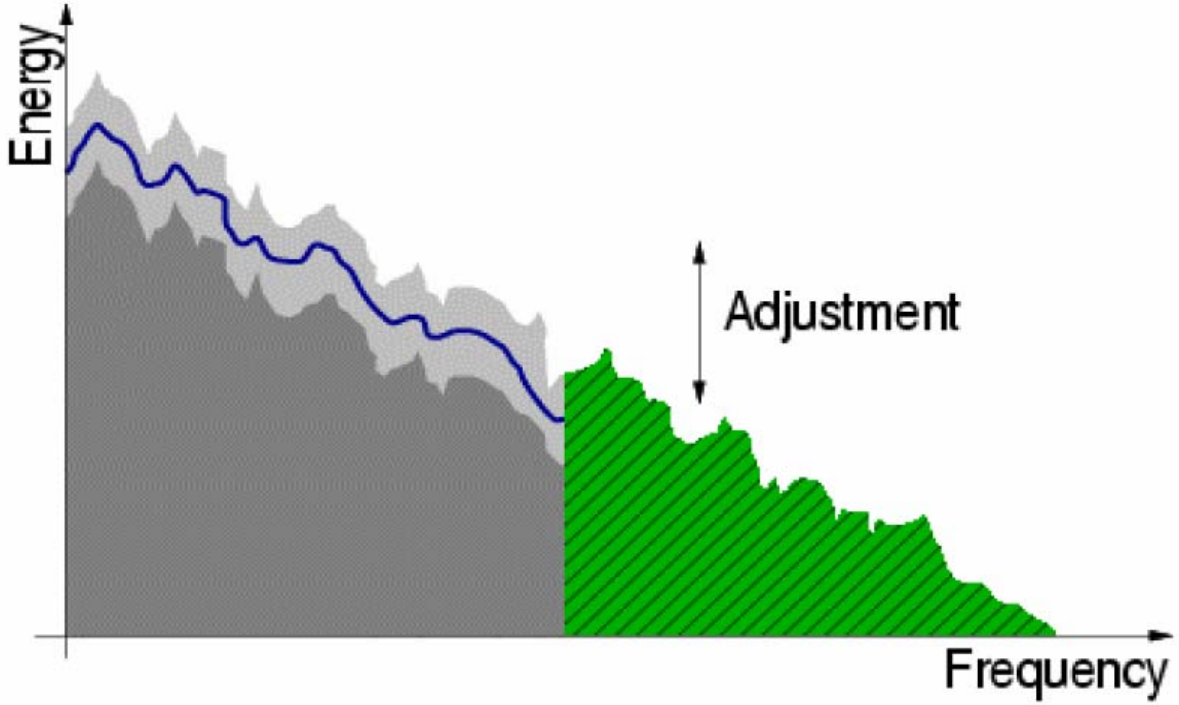
SBR



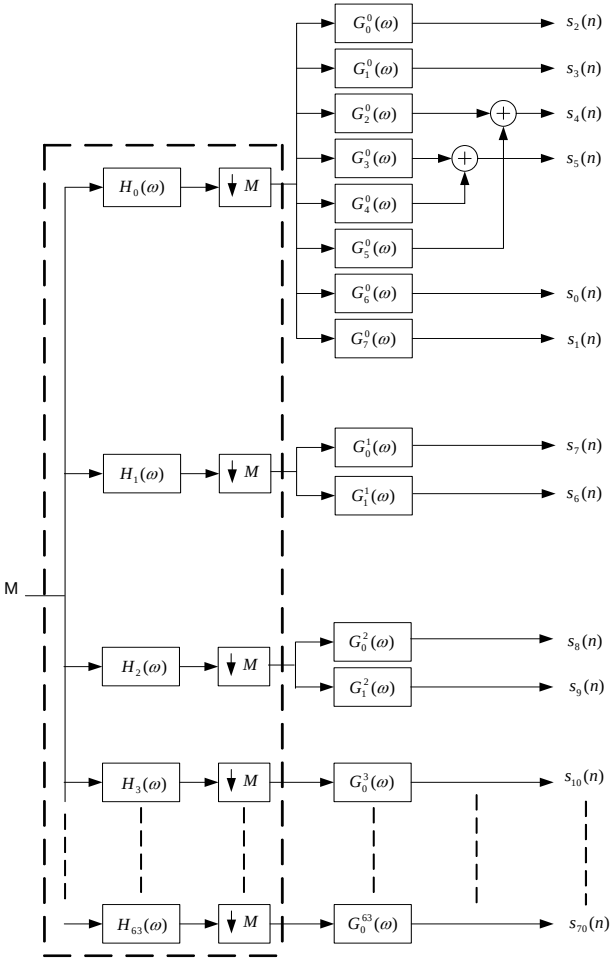
SBR

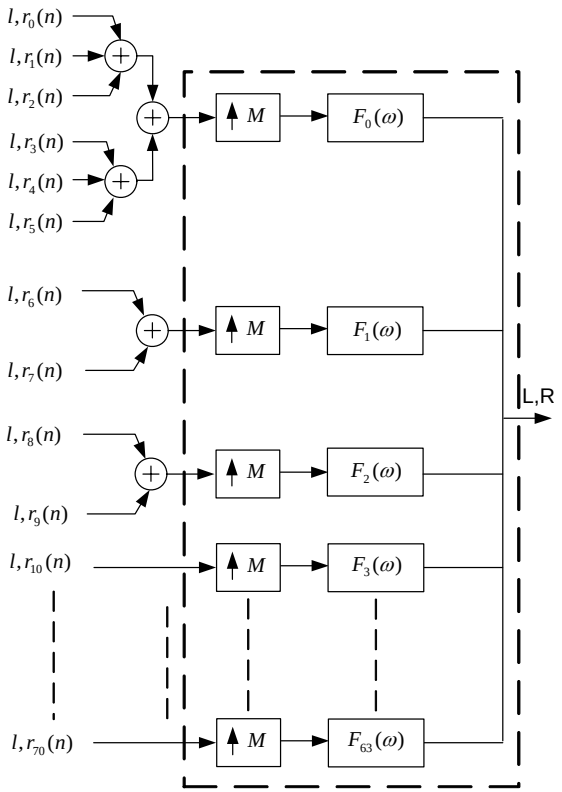


SBR



PS


M
$H_0(\omega)$
$H_1(\omega)$
$H_2(\omega)$
$H_3(\omega)$
$H_{63}(\omega)$
M
$G_0^0(\omega)$
$G_1^0(\omega)$
$G_2^0(\omega)$
$G_3^0(\omega)$
$G_4^0(\omega)$
$G_5^0(\omega)$
$G_6^0(\omega)$
$G_7^0(\omega)$
$G_0^1(\omega)$
$G_1^1(\omega)$
$G_0^2(\omega)$
$G_1^2(\omega)$
$G_0^3(\omega)$
$G_0^{63}(\omega)$
$s_2(n)$
$s_3(n)$
$s_4(n)$
$s_5(n)$
$s_0(n)$
$s_1(n)$
$s_7(n)$
$s_6(n)$
$s_8(n)$
$s_9(n)$
$s_{10}(n)$
$s_{70}(n)$


$l, r_0(n)$
$l, r_1(n)$
$l, r_2(n)$
$l, r_3(n)$
$l, r_4(n)$
$l, r_5(n)$
$l, r_6(n)$
$l, r_7(n)$
$l, r_8(n)$
$l, r_9(n)$
$l, r_{10}(n)$
$l, r_{70}(n)$
M
$F_0(\omega)$
$F_1(\omega)$
$F_2(\omega)$
$F_3(\omega)$
$F_{63}(\omega)$
L,R