Media Processing Challenges

- Increasing performance requirements
- Need for flexibility & scalability
- Increasing cost and time-to-market pressure
- Need to provide system solution
  - System software very different from media processing software
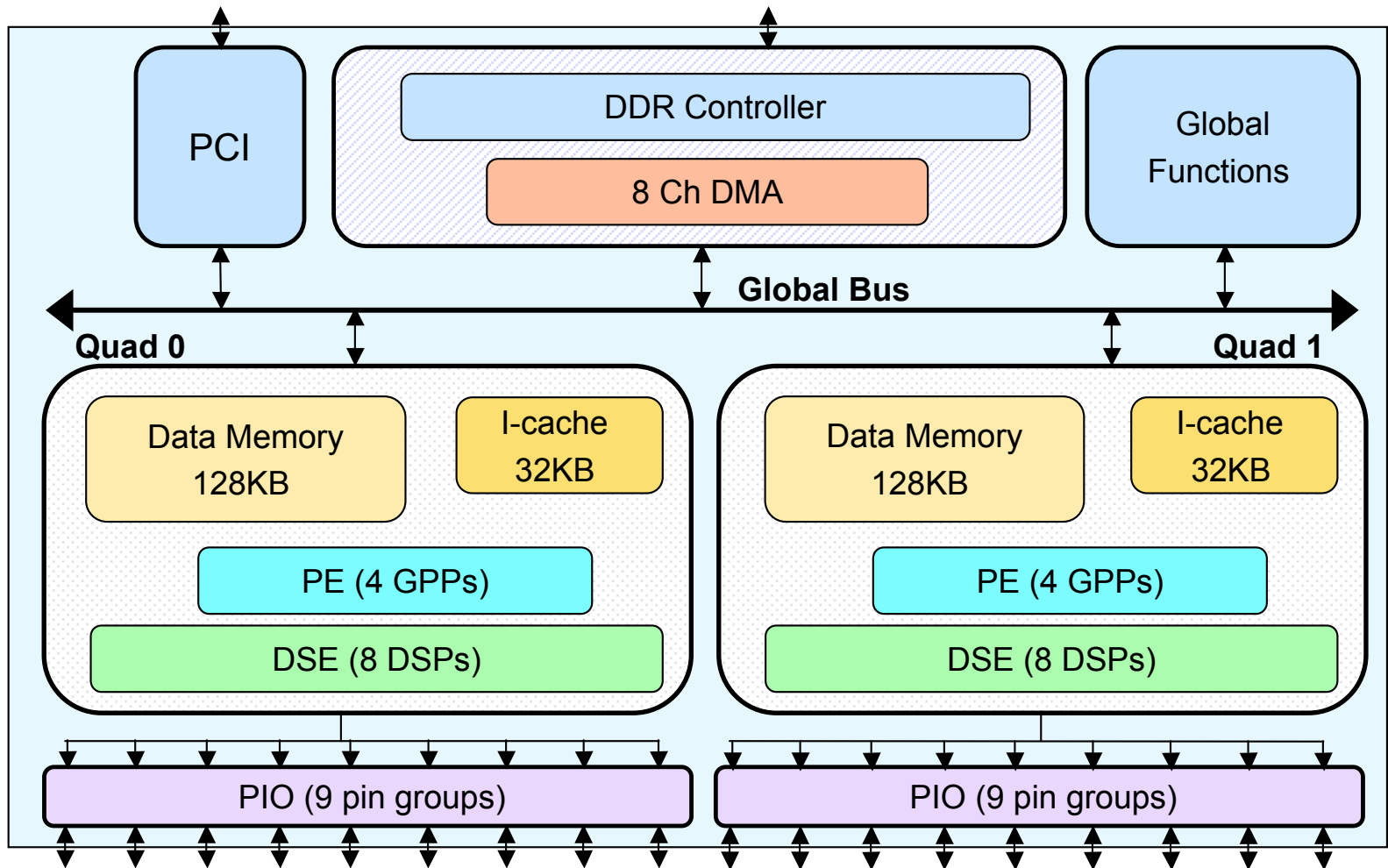 - Need to provide required I/O

New Shift to Thread-Parallel Computing

- VLIW alone does not solve all these challenges
- Thread-level parallel architectures are becoming viable
  - Increased transistor budgets at newer process technologies
  - Multithreaded programming model and tools maturing
- Reduced power achieved versus complex, single-core processors
- Industry is switching to thread-parallel models
  - Cell processor, XBox 360
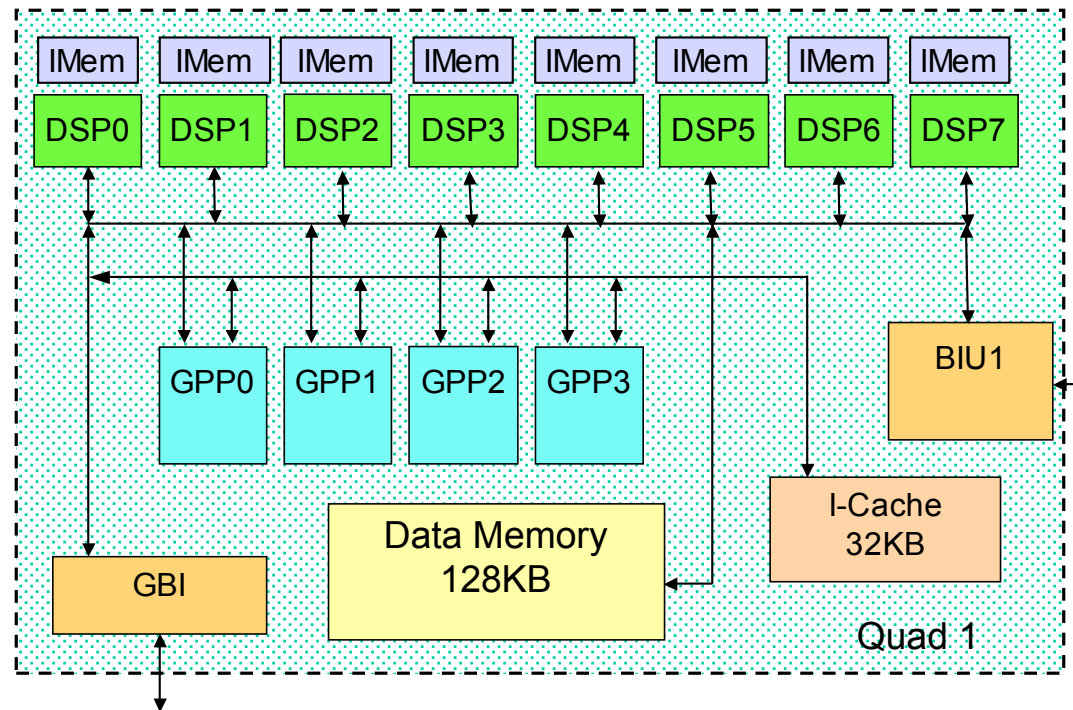 - Multi-core roadmaps emerging everywhere

MDSP Solution

- Multiprocessor DSP (MDSP) architecture addresses media processing challenges
  - Performance
    - Multiple area-efficient engines => higher true MIPS / mm^2
    - Scalable memory hierarchy for high bandwidth
  - Scalability
    - Can scale performance by number of cores, not just frequency
    - Loosely coupled programming model makes it easy to scale software as number of cores changes
  - Ease of optimization
    - Dedicated IMem => No ICache misses
    - Threads with single instruction per cycle simplifies optimization
  - System solution
    - GPP used for control code so DSP can be kept busy with computation
    - Multiple processors => multiple control threads => improved real-time behavior
    - Programmable I/O provides user-defined mix of I/O interfaces

CT3616 Block Diagram

MDSP ADVANTAGE

CRADLE

PCI

DDR Controller

8 Ch DMA

Global Functions

Global Bus

Quad 0

Quad 1

Data Memory 128KB

I-cache 32KB

Data Memory 128KB

I-cache 32KB

PE (4 GPPs)

PE (4 GPPs)

DSE (8 DSPs)

DSE (8 DSPs)

PIO (9 pin groups)

PIO (9 pin groups)

Quad Structure

IMem	IMem	IMem	IMem	IMem	IMem	IMem	IMem
DSP0	DSP1	DSP2	DSP3	DSP4	DSP5	DSP6	DSP7

GPP0　GPP1　GPP2　GPP3

BIU1

Data Memory
128KB

I-Cache
32KB

GBI

Quad 1

- 2 Compute subsystems with:
  - 8 DSP engines
  - 4 General Purpose Processors
  - 128 KB of shared data memory
  - 32 KB of shared instruction cache
  - Separate instruction & data buses

- Close I/O subsystem integration
  - Integrated bus interface unit
  - Efficient input stream pre-processing
  - Reduce DRAM accesses

DSP Cores
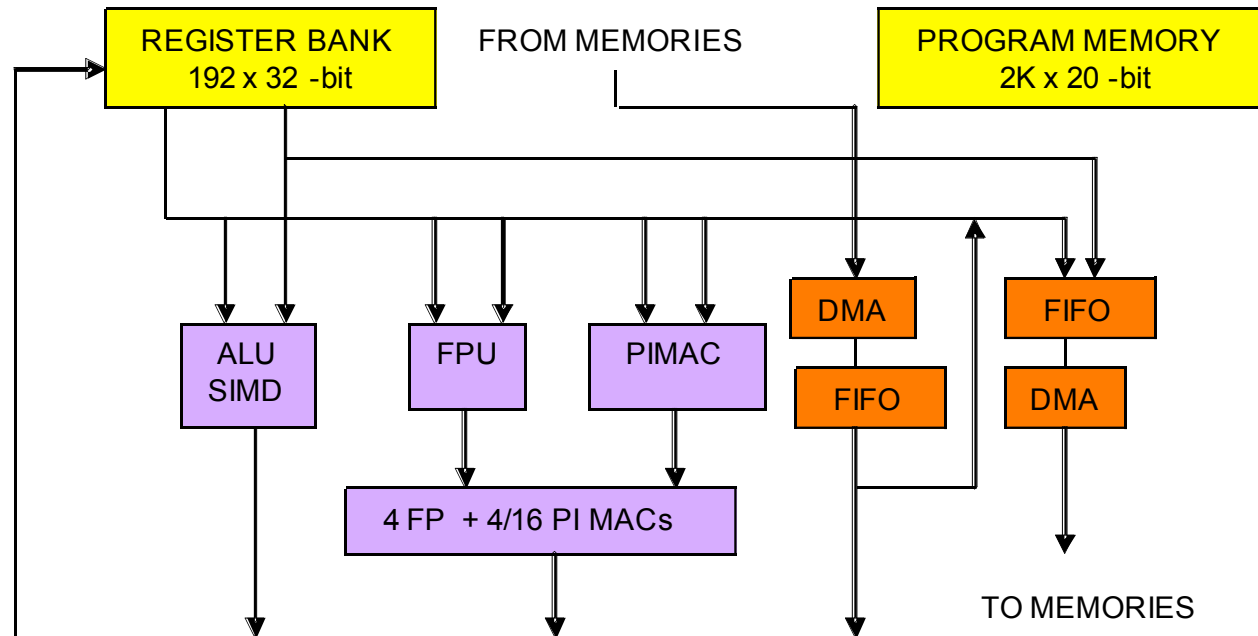
- High memory bandwidth provided by 192 registers with background transfers to/from local data memory
- MAC with 32 bit floating point or 4x8-bit, 2x16-bit, or 1x32-bit integer operands
- 16 parallel signed/unsigned 8x8 bit MAC operations per clock (256 MACs/cycle)
- SIMD operations on packed 8-bit or 16-bit data
- Sum of Absolute Differences (SAD) for video encoding
- Zero-cycle arbitrary bit-field access and packing

Memory Architecture offers High Bandwidth
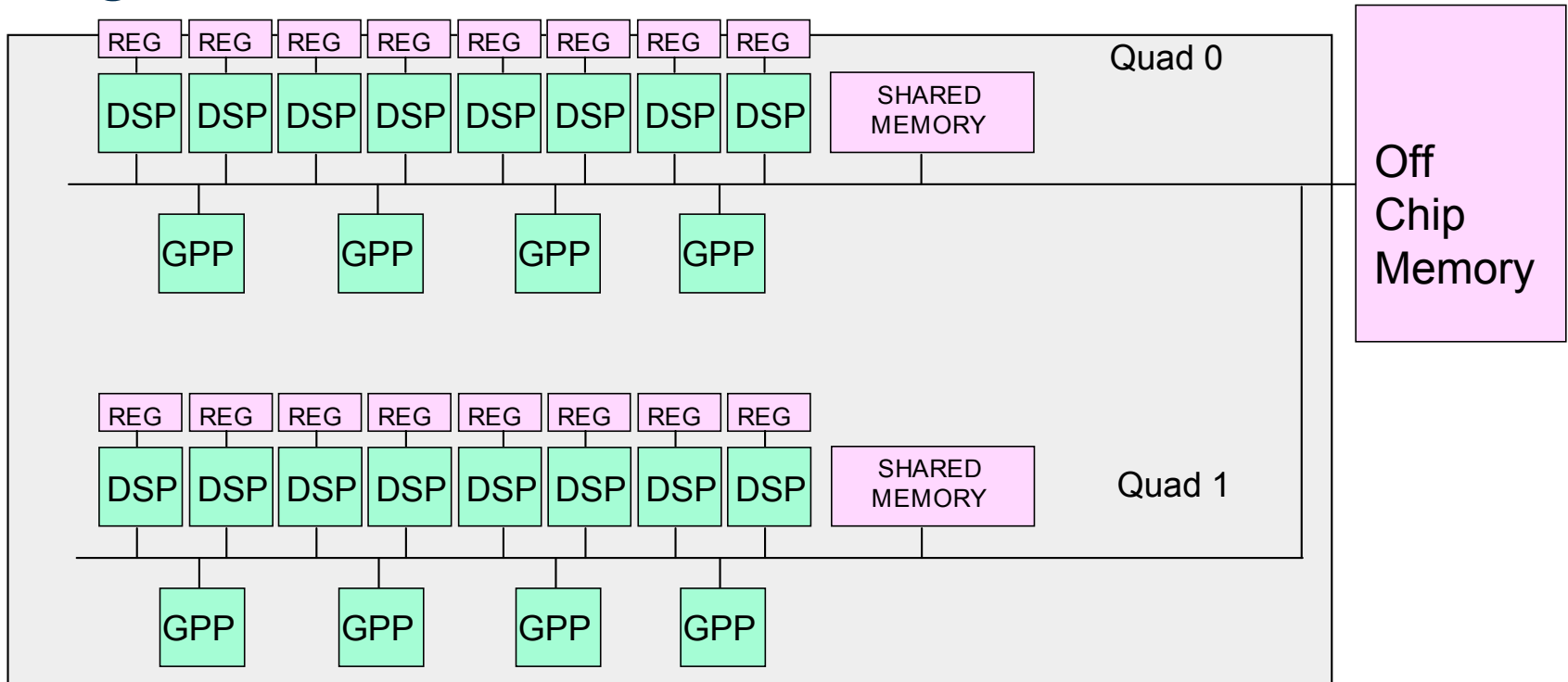
- Three-level memory & bus hierarchy (90% of accesses are local DSP mem)
- At 350 MHz DSP clock, 333 MHz DRAM clock:
  - 16 DSPs each 3 (2R, 1W) x 4 bytes x 350 MHz bandwidth to registers   (67 GB / s)
  - 2 shared memories each with 8 bytes x 350 MHz bandwidth   (5.6 GB / s)
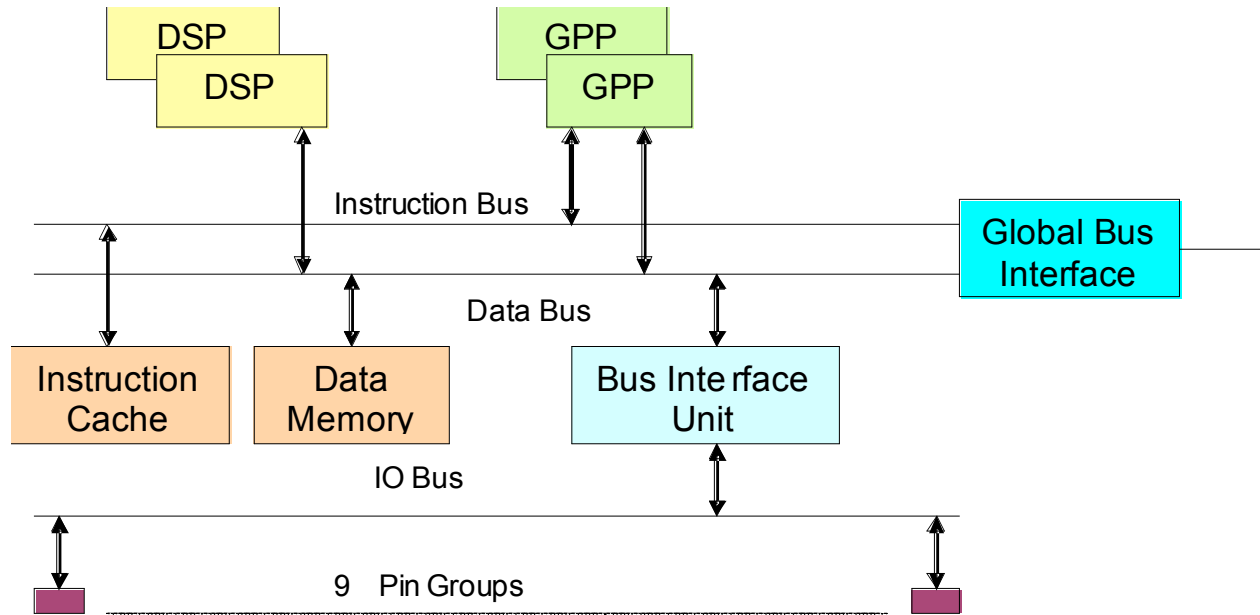 - 1 DDR DRAM with 8 bytes x 333 MHz bandwidth   (2.7 GB / s)

DRAM Performance
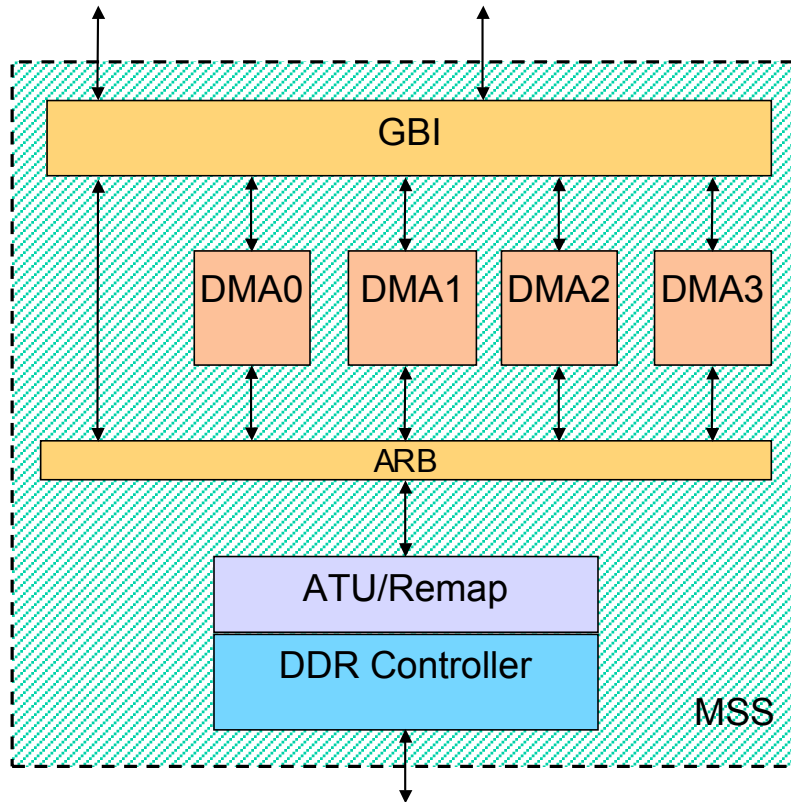
- DDR 333 MHz data rate

- GPP/DSP DRAM writes fast

- 4 DMA engines
    - DRAM reads fast
    - Pre-fetch 1D and 2D data from DRAM to shared memory
    - Support chained transfers

- Memory remapping
    - Minimize page misses for 2D

I/O Subsystem

- BIU transfers data from/to pin-groups (which do pin level signaling interfaces)
- Scatter/Gather HW in BIU, streams data directly between memories and BIU
- DSPs do ECC checking (CRC, Reed-Solomon) & low-level command/status/data processing
- DSPs can also do preprocessing of data (downsampling, scaling, etc.)
- GPPs support Device level APIs for command/status/data high-level interfaces.

Pin Groups
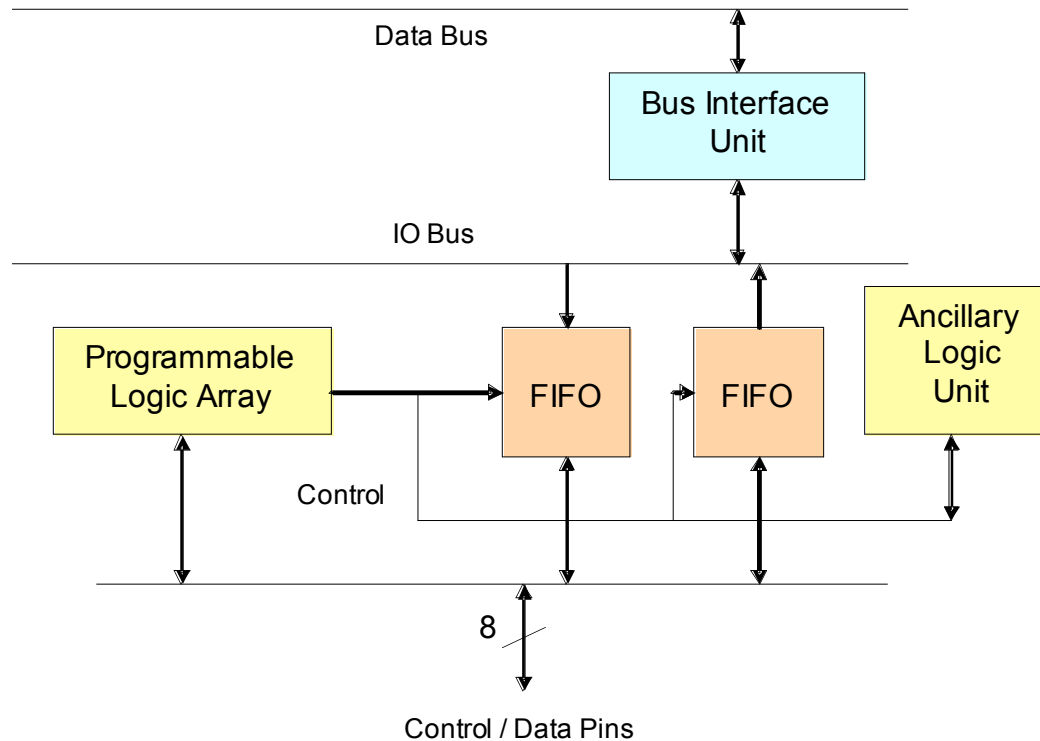
- 18 pin groups (CT3616) of 8 pins each
- The PLA provides a fast state machine for implementing pin level signaling interfaces
- FIFOs and serial I/O units assemble words
- Ancillary logic has commonly used units, e.g. timers, parity, comparators
- Pin Groups can be grouped together for wider devices

CT3600 Programming Model

- Two main types of parallelism
  - Data parallelism – each processor works on own chunk of data
  - Pipelined parallelism – each processor does part of the algorithm on a chunk of data, then passes chunk on to next stage in pipeline
  - Most applications have some of both
- Loosely coupled, coarse-grain multiprocessing
  - Tasks running on different processors process large chunks of data, independently
  - Synchronization and sharing of data is small percentage of work
  - Media applications fit this model very well
- Master / Slave model
  - GPPs do control code and synchronization (masters)
  - DSPs do heavy computation and processing (slaves)
  - Each processor can be optimized for respective type of code => efficient use of silicon

Dynamic Scheduling

- ## Dynamic Scheduling of Tasks
  - Two classes of tasks: GPP tasks and DSP tasks
  - Algorithm is statically partitioned into these two classes
  - One or more software queues for each class
  - When processors finish their current task, they grab next task from the appropriate queue
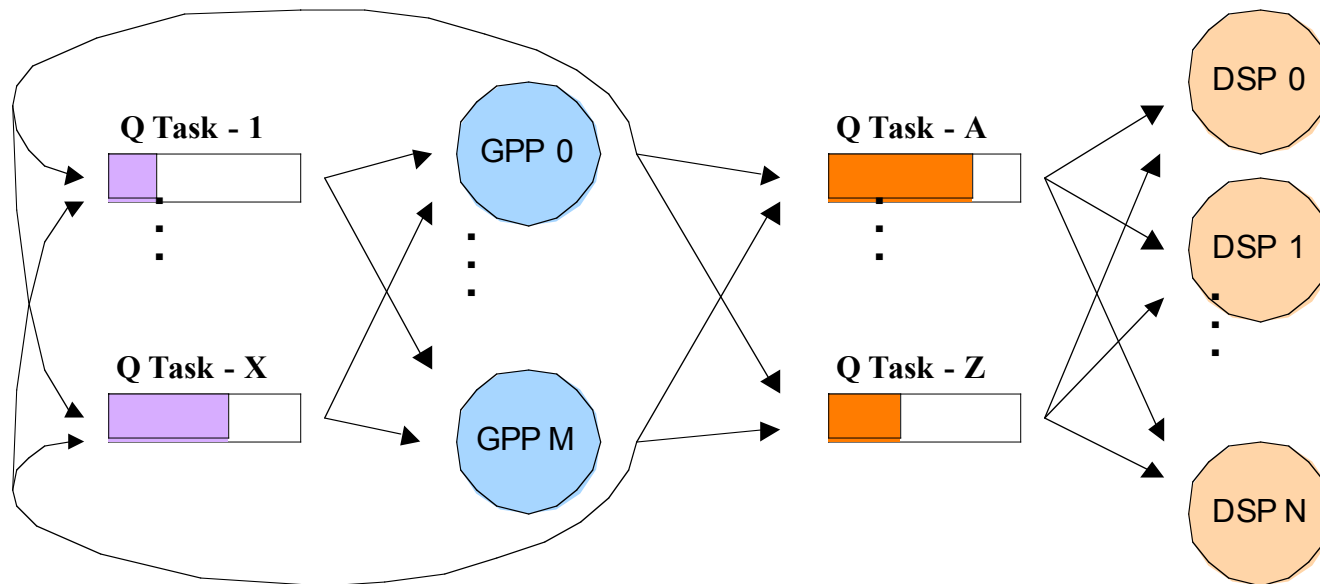
Dynamic Scheduling (Cont'd)

- ## Advantages of Dynamic Scheduling
  - Easier to partition – Only need to worry about partitioning tasks into GPP or DSP task, not about which GPP or DSP to run it on
  - Scalable – Same application can run with any number of GPP or DSP resources to get desired performance
  - Automatic load balancing – Even with software pipelining, no processor is idle, since they grab next task as soon as they finish
  - Low overhead – For H.264 encoder, overhead is < 2%

INSPECTOR™
Graphical multiprocessor debugger

Unlimited Breakpoints Across Multiple Processors

Symbolic Debugging Support

Debug Multiple Processors at Once

View Multiple Processor Registers

Processor Status Summary

Multiprocessor Profiling and Performance Analysis Tools

Customized Profiling

Identify Performance Bottlenecks Quickly

Function and Instruction Profiling
True Multiprocessing Support

Performance

	TI DM642™	Cradle CT3616™
DSP Cores	1	16
GPP Cores	0	8
Freq (MHz)	600	350
16-bit MMACs / sec	2400	22400
MPEG4 Channels *	4	16
$ / Channel **	$10	$5.50
mW / Channel	450	300

* Channel: MPEG4 SP L3, CIF Encoder @ 30fps
** Based on 10K book price

Performance (cont'd)

CODEC	Resolution	% of 3616 Used for Encode
MJPEG	D1 @ 30 FPS	8 %
MPEG4	CIF @ 30 FPS SP	6 %
MPEG4	D1 @ 30 FPS SP	22 %
H.264	CIF @ 30 FPS MP	17 %
H.264	D1 @ 30 FPS MP	75 %

MDSP ADVANTAGE

CRADLE
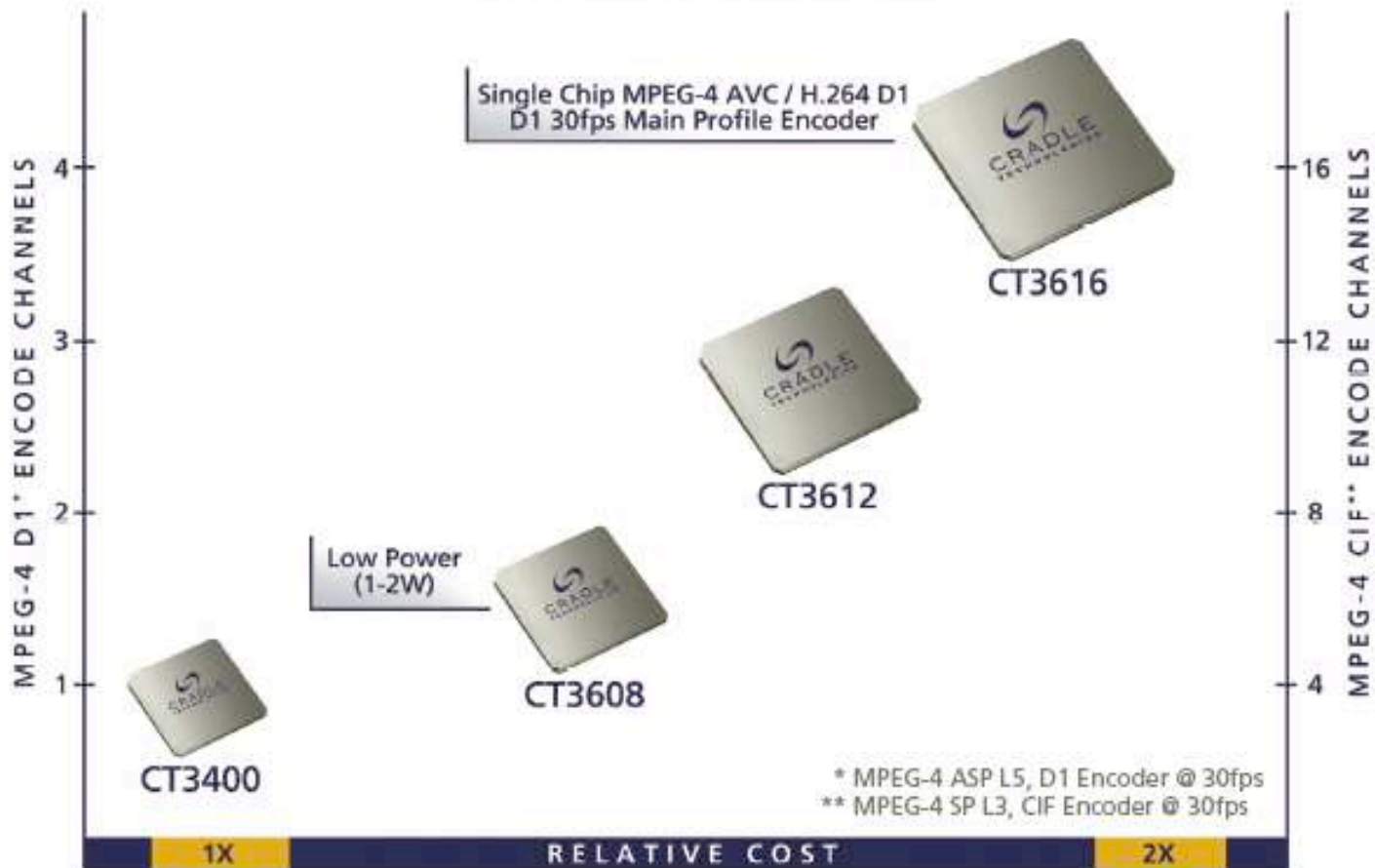
CT3616 Silicon Status

- TSMC 0.13 µm LV process
- 55 million transistors
- < 5 W @ 1.2 V, 350 MHz, running DV stress test
- 400 KB on-chip RAM/Cache
- First silicon received 1Q 05
- General samples with development board and BSP available in 3Q 05

CT3600 Family

Scalable Price / Performance

Single Chip MPEG-4 AVC / H.264 D1
D1 30fps Main Profile Encoder

CT3616

CT3612

Low Power
(1-2W)

CT3608

CT3400

MPEG-4 D1* ENCODE CHANNELS

MPEG-4 CIF** ENCODE CHANNELS

* MPEG-4 ASP L5, D1 Encoder @ 30fps
** MPEG-4 SP L3, CIF Encoder @ 30fps

1X   RELATIVE COST   2X

Summary

- Multicore Architecture for Media Processing
  - 16 DSP cores optimized for video/media: SIMD, SAD, PIMAC with up to 89.6 GMACs (8x8)
  - 8 GPP cores for efficient handling of control code
  - High data throughput: DDR, DMAs, high-speed internal bus
  - Flexible: programmable cores, programmable I/O
  - Scalable programming model to simplify design
- State-of-the-Art Development Tools
  - Graphical multiprocessor debugger/profiler
  - Multiprocessor dynamic scheduling and run time analysis
- High Performance
  - 1 H.264 D1, or 16 MPEG4 CIF channels, encode
  - With resources to spare for audio, intelligent video, I/O, etc. => True single-chip solution

Thank you

For more information, or to schedule a demo

visit www.cradle.com

or contact sales@cradle.com