Chapter 5-1: Multiprocessor Architectures

Soo-Ik Chae

Topics

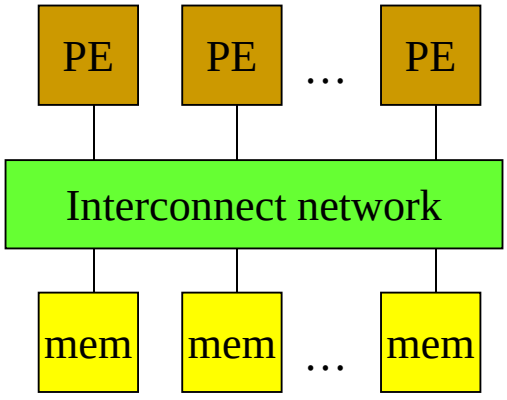
- Motivation.
- Architectures for embedded multiprocessing.
- Interconnection networks.

Motivation

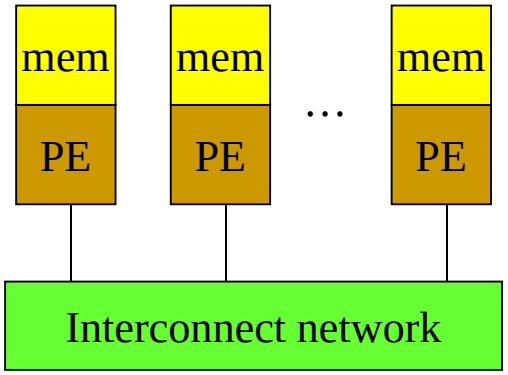
- Multiprocessing is very common in embedded computing systems because it allows us to meet our performance, cost, energy/power consumption goals.
- Often heterogeneous multiprocessors
- Software must be carefully designed to obtain the most out of the multiprocessor
- A multiprocessor system consists of three major components
  - Processing elements that operate on data
  - Memory blocks that hold data
  - Interconnection networks of processing elements and memory

Generic multiprocessor

- Shared memory:



- Message passing:



Design choices

- Processing elements:
  - Number.
  - Type.
  - Homogeneous or heterogeneous.
- Memory:
  - Size.
  - Private or shared memories.
- Interconnection networks:
  - Topology.
  - Protocol.

Why embedded multiprocessors?

- Real-time performance---segregate tasks to improve predictability and performance.
  - A combination of hardware and software must be used to provide predictable performance
- Low power/energy---segregate tasks to allow idling, segregate memory traffic.
  - Low power: heating
  - Low energy: battery life
- Cost---several small processors are more efficient than one large processor.
  - Must provide high performance with less hardware

Example: cell phones

- Variety of tasks:
  - Error detection and correction.
  - Voice compression/decompression.
  - Protocol processing.
  - Position sensing.
  - Music.
  - Cameras.
  - Web browsing.

Design techniques

- The rigorous demands of embedded computing push us toward several design techniques
  - Heterogeneous multiprocessors are often energy-efficient and cost-effective than symmetric multiprocessors.
  - Heterogeneous memory systems improve real-time performance
  - Networks-on-chips support heterogeneous architecture.

Example: cellular phones

- A cellular phone must perform a variety of functions that are basic to telephony.
  - Compute and check error-correction codes.
  - Perform voice compression and decompression
  - Respond to the protocol that governs communication with the cellular network.
- Modern cell phones must perform advanced functions that are either required by regulation or demanded by the marketplace.
  - A global positioning system (GPS) function: cell phones in the US must be located for emergency services.
  - Many cell phones play MP3 audio. They may also use MIDI to play music for ring tones
  - High-end cell phones provides cameras for still pictures and video.
  - Cell phones may download application code form the network.

Example: video cameras

- Three basic methods to compress video
  - Lossless compression
    - VLC
    - CABAC
  - DCT to help quantize the images and reduce the size of the video stream by lossy encoding.
  - Motion estimation and compensation
- Of these three, ME is the most computationally intensive task.
  - Per macroblocks of 16 x 16.
  - For the search range

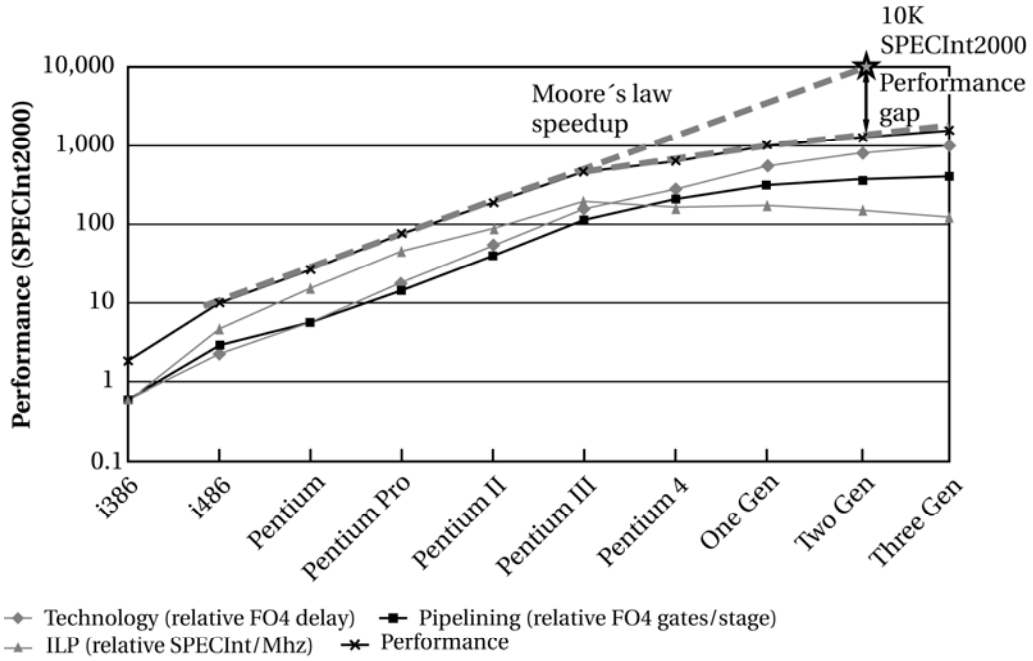
Example: video compression

- QCIF (176 x 144) used in cell phones and portable devices:
  - 99 macroblocks of 16 x 16 per frame.
  - Frame rate of 15 or 30 frames/sec.
  - Assuming seven correlations per macroblock, 7x16x16x99 absolute difference computations per frame.
  - A DCT algorithm uses 94 multiplications and 454 additions per 8 x 8 2D DCT
    - (곱셈, 덧셈) = (94, 454) x 99 x 6 / frame

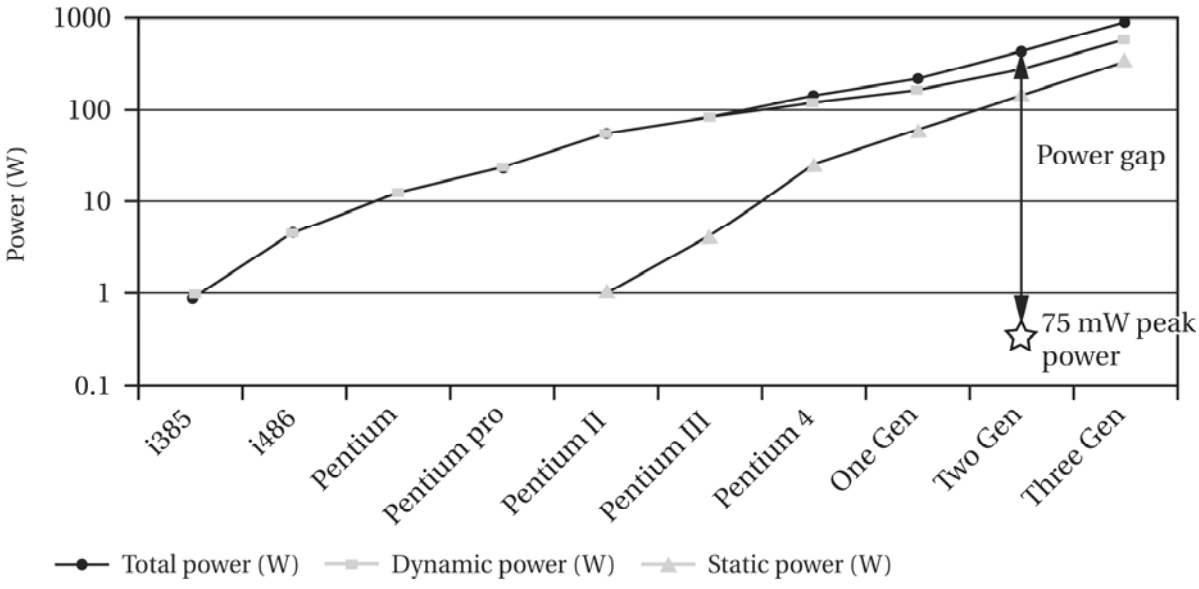
Performance and energy

- Next-generation workload on portable device:
  - Speech compression.
  - Video compression and analysis.
  - High-resolution graphics.
  - High-bandwidth wireless communications.
- Workload is 10,000 SPECint
  - About 16 x 2GHz Pentium 4.
- Battery must consume no more than 75 mW.
  - Based on the assumption of using a 2006 battery for 5 days
  - Battery power is growing at only 5% per year

Performance trends on desktop



Energy trends on desktop



Task-level parallelism

- Many embedded applications can be divided into several tasks that communicate each other.
- Desktop processors relay on instruction-level parallelism to improve performance.
  - Only a limited amount of instruction-level parallelism is available in most programs
- We can build custom multiprocessor architectures that reflect the task-level parallelism and meet performance targets at much power lower cost and with much less energy.

Multiprocessing

- Multiprocessors are driven by
  - High performance
  - Low power
  - Real time.
- It is cheaper to split the system functions over several processors.
  - The manufacturing cost of a microprocessor is a super-linear function of clock speed.
- Real time requirement leads us to multiprocessing
  - Several tasks run on the same processor compete for cycles.
  - We cannot use 100% of the CPU if we want to guarantee that we can met the deadlines.
  - Furthermore, we must pay for those reserved cycles at the super-linear rate of the higher clock speed.

Specialization and multiprocessing

- Heterogeneity reduces power consumption.
- Heterogeneity improves real-time performance. (less intuitively)
- Why use heterogeneous multiprocessors:
  - Some operations (8 x 8 DCT) are standardized.
  - Some operations that do not map well on a CPU need to be specialized:
    - Bit-level function
    - Too many registers
    - Controlling the precision of the arithmetic.
  - Highly responsive I/O operations may be best performed by an accelerator with an attached I/O unit.

Heterogeneity

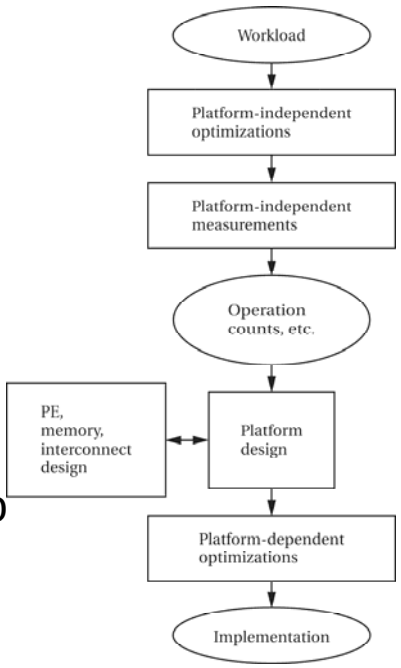
- Heterogeneity reduces power consumption because it removes unnecessary hardware.
- Excessive specialization can add so much communication cost that the energy gain from specialization is lost.
  - Specializing the right functions can lead to significant energy savings.
- We can often meet deadlines and be responsive to interaction much more easily when we put those time-critical processes on separate CPUs.
- Specialized memory systems and interconnections also help make the response time of a process more predictable.

Flexibility and efficiency

- Why programmable processors?
- Many embedded systems perform complex functions that would be too difficult to implement entirely in hardware.
- This is particularly true of standards-based systems.
- Standards is often represented as a reference implementation in a standard programming language.
- Translating all the standards to hardware may be too time-consuming and expensive.
- Multiple standards encourage software implementation
- Given the hundreds of thousands of lines of code that must be implemented, processors running software, perhaps aided by a few key hardware units, are the only reasonable design choices.

Multiprocessor design methodologies

- Analyze workload that represents application's usage.
  - In contrast to bentchmarks
- Platform-independent optimizations eliminate side effects due to reference software implementation.
- Platform design is based on operations, memory, etc.
- (Platform-dependent optimization) Software can be further optimized to take advantage of platform.



Cai and Gajski modeling levels

- Implementation: corresponds directly to hardware.
  - Captures both computation and communication to cycle accuracy
- Cycle-accurate computation:
  - captures accurate computation times
  - approximate communication times.
- Cycle-accurate communication:
  - captures communication times accurately
  - but computation times only approximately.
- Bus-transaction: models the basic function of a bus arbitration sceme
  - models bus operations but is not cycle-accurate.
- PE-assembly: represent the system as PE communicating through the channel
  - communication is untimed,
  - PE execution is approximately timed.
- Specification: functional model.

Cai and Gajski modeling methods

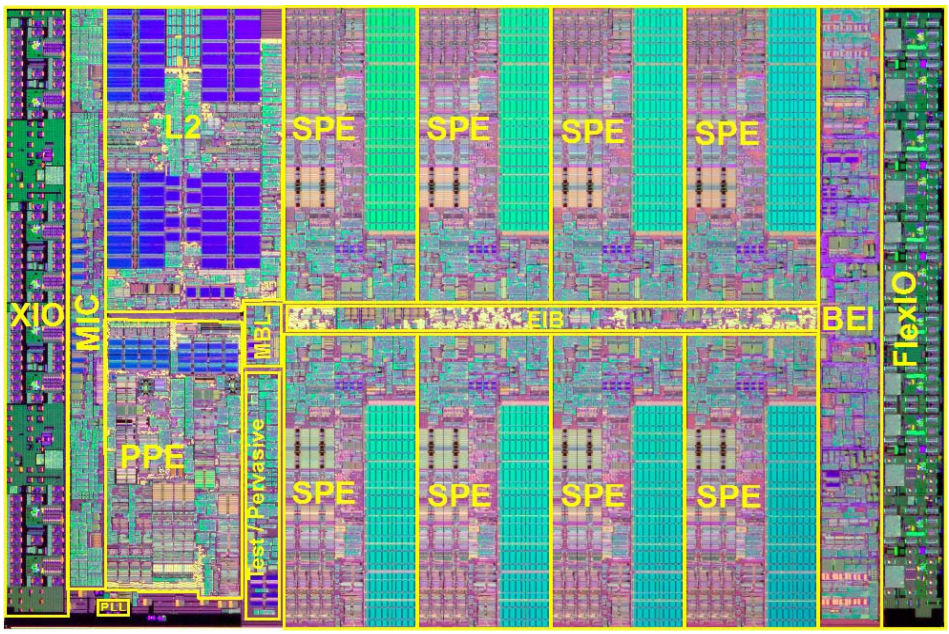
	Communication time	**Computation time**	**Communication scheme**	**PE interface**
Spec	No	No	Variable	No PEs
Component assembly	No	Approximate	Variable channel	Abstract
Bus arbitration	Approximate	Approximate	Abstract bus channel	Abstract
Bus functional	Cycle accurate	Approximate	Protocol bus channel	Abstract
Cycle accurate	Approximate	Cycle accurate	Abstract bus channel	Pin accurate
Implementation	Cycle accurate	Cycle accurate	Wires	Pin accurate

Multiprocessor systems-on-chips

- MPSoC is a complete platform for an application.
- Generally heterogeneous processing elements.
- Combine off-chip bulk memory with on-chip specialized memory.

1. Cell Processor

• 90nm 8 level metal CMOS SOI technology
• effective gate length: 45 nm (NMOS) 60nm (PMOS)
• area: 231 mm^2



Cell Processor Architecture

(1) embedded processors:

1 Power core (PPE), 8 Synergistic Processing Elements (SPEs)

(2) processing engine (PE)

No function specific PE.

(3) on-chip memory

32KB L1 data and instruction caches in Power core

512KB L2 cache in PPE, 256KB local store in each SPE

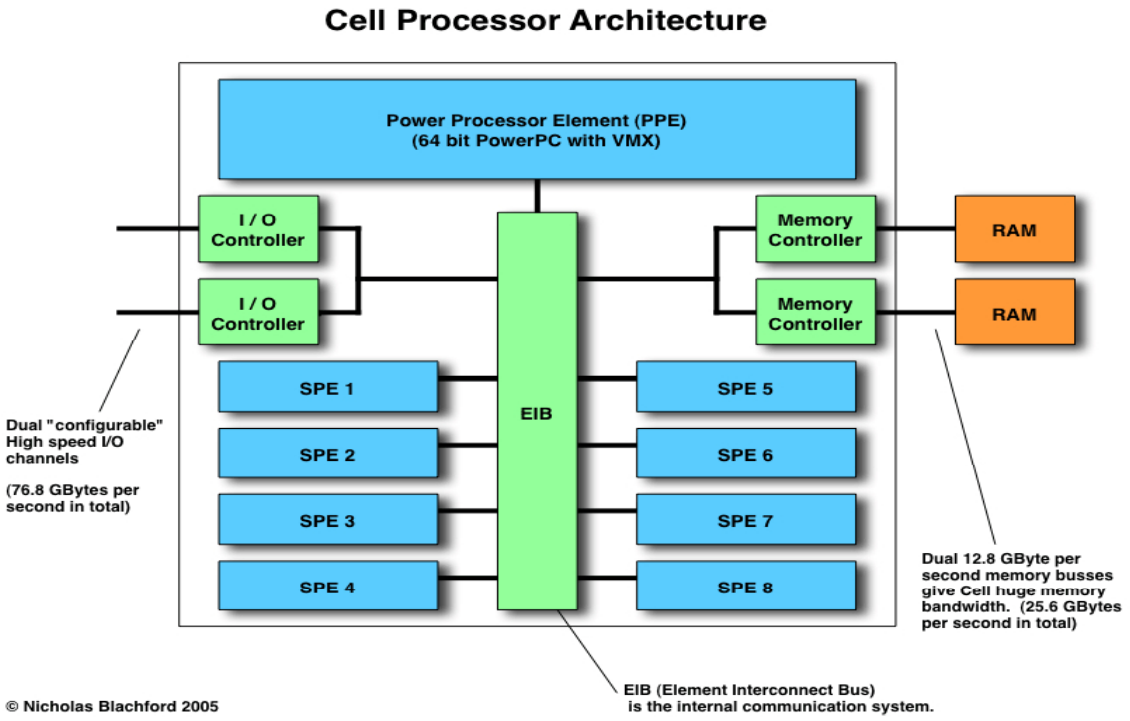
(4) off-chip memory

RAMBUS XDR DRAM

(5) i/o device controllers (peripherals)

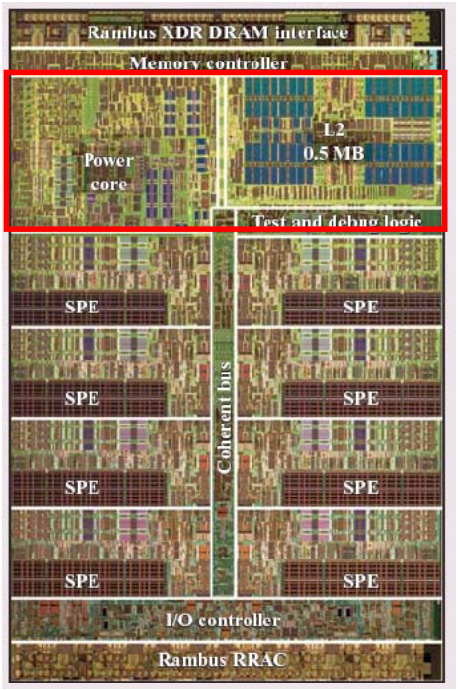
(6) interconnection structure

Cell Processor Architecture

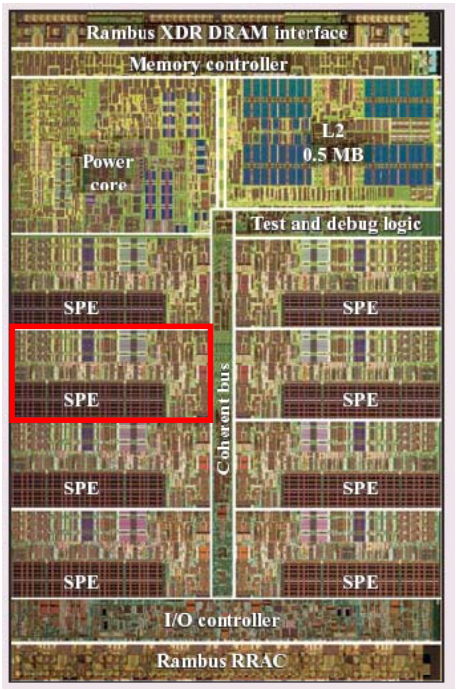


Power Processor Element (PPE)

- A 64 bit, "Power Architecture" processor with 32KB L1 data and instruction caches
- 512KB L2 cache.
- 11 FO4 design
- A dual issue, dual threaded, in-order processor.
- SIMD unit for media processing



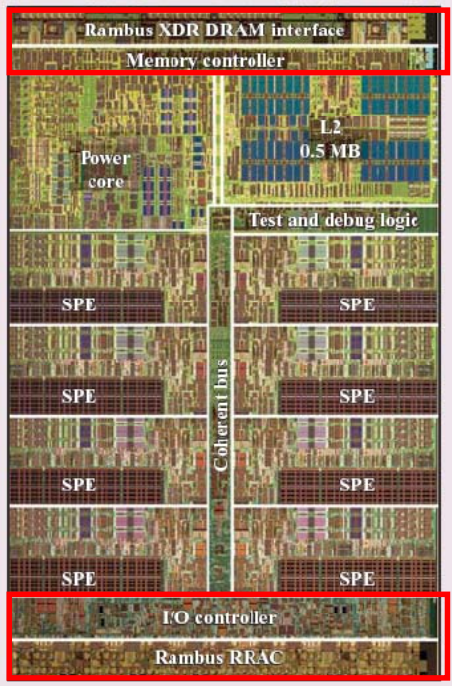
Synergistic Processor Elements (SPEs)



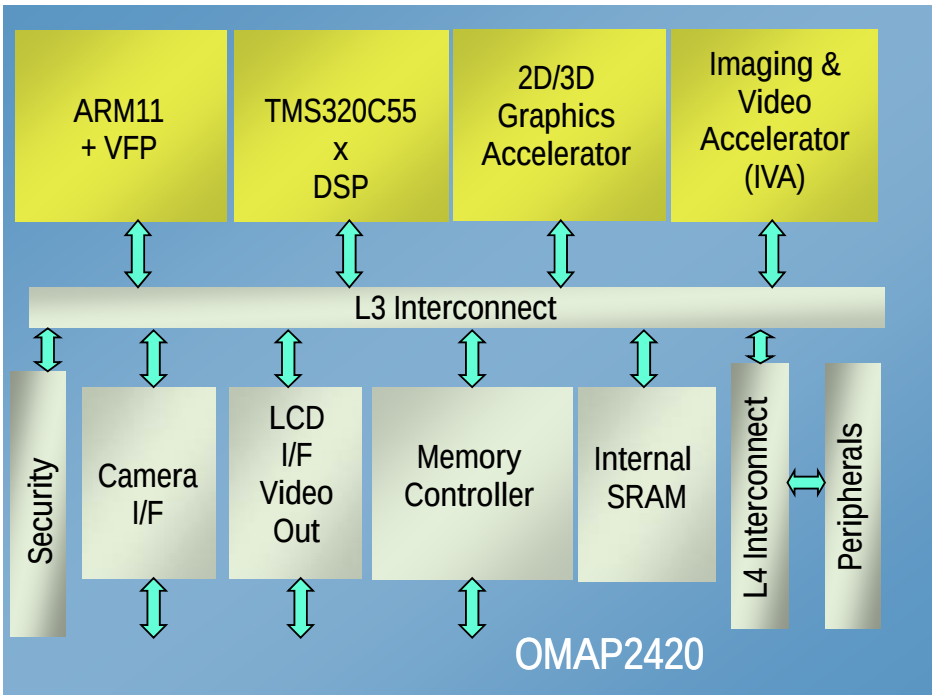
- An SPE is a self contained vector processor which acts as an independent processor.
- Each SPE contains
  - 128 x 128 bit registers
  - 4 single precision FPUs capable of 32 GigaFLOPS at 4GHz
  - 4 Integer units capable of 32 GOPS at 4GHz.
  - 256 KB local store instead of cache
- Area: 15 mm^2
- Consumes less than 5 Watts at 4GHz
- perform as well as a top end (single core) desktop CPU given the right task

Memory and I/O Controller

- memory controller
  - Rambus XDRAM interface to Rambus XDR memory
  - dual channels at 12.8 GBps → 25.6 GBps
- I/O controller
  - Rambus FlexIO interface which can be clocked independently
  - dual configurable channels
  - maximum ~ 76.8 GBps



2. OMAP 2420 Processor (TI)



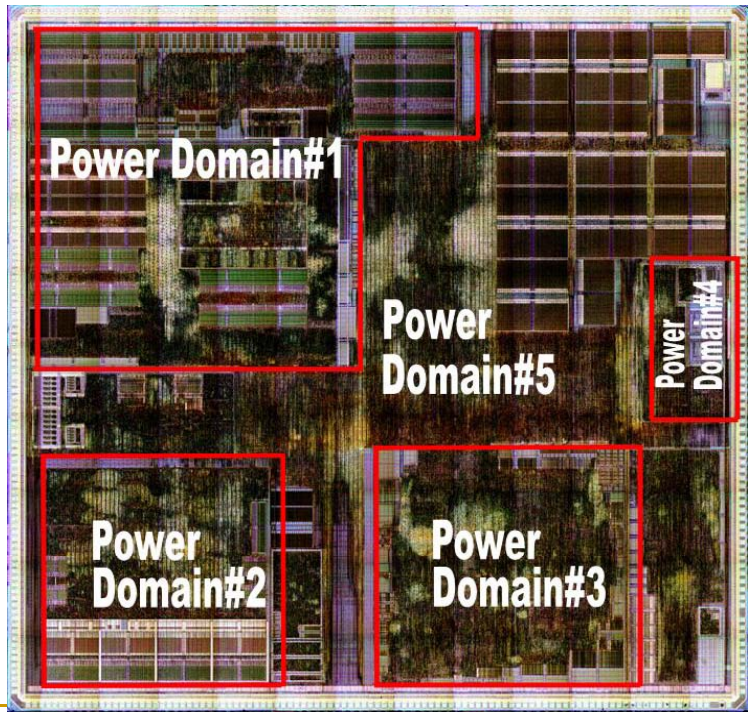
OMAP 2420 Features

OMAP 2420 processor contains

- ARM 1136 @ 330 MHz
  VFP (Vector Floating Point)
  32KB Icache and 32KB Dcache
- C55 DSP @ 220 MHz
- 2D/3D graphics accelerator
- IVA decodes
  still images to >4 Mpixels
  30 fps VGA video decode
- Output to TV for gaming and video playback
- Encryption hardware for DRM and security

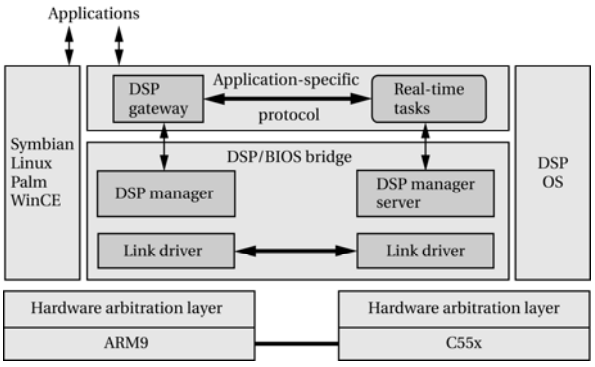
OMAP2420

5 Power Domains
- #1: ARM11 Core
- #2: C55 DSP Core
- #3: Graphic Accelerator
- #4: Always On logic
- #5: Connect + Periph.



TI OMAP

- Designed for mobile multimedia.
- C55x DSP performs signal processing as slave.
- ARM runs operating system, dispatches tasks to DSP.



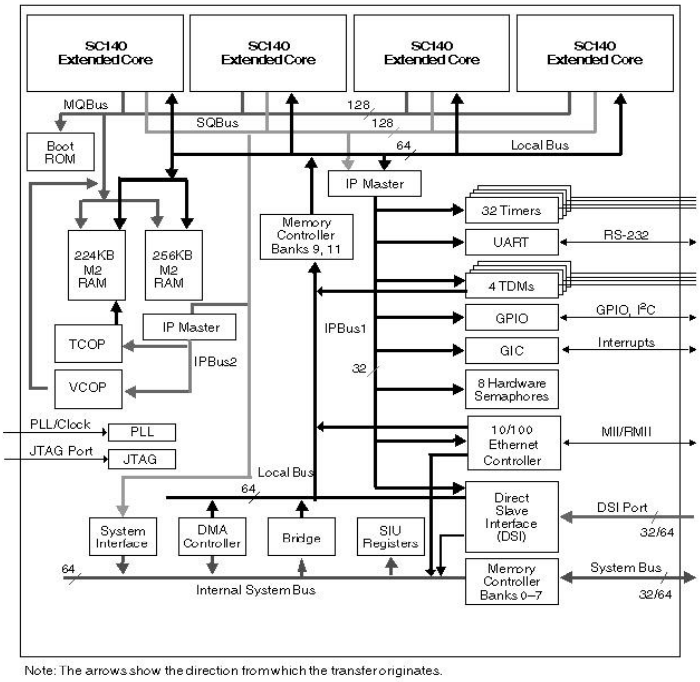
3. MSC8126 (Freescale)

Four 400/500 MHz StarCore SC140 DSP extended cores

Internal DMA controller, enabling data transfers of to and from the SC140 core, M1 memory, the M2 memory, and the serial interfaces

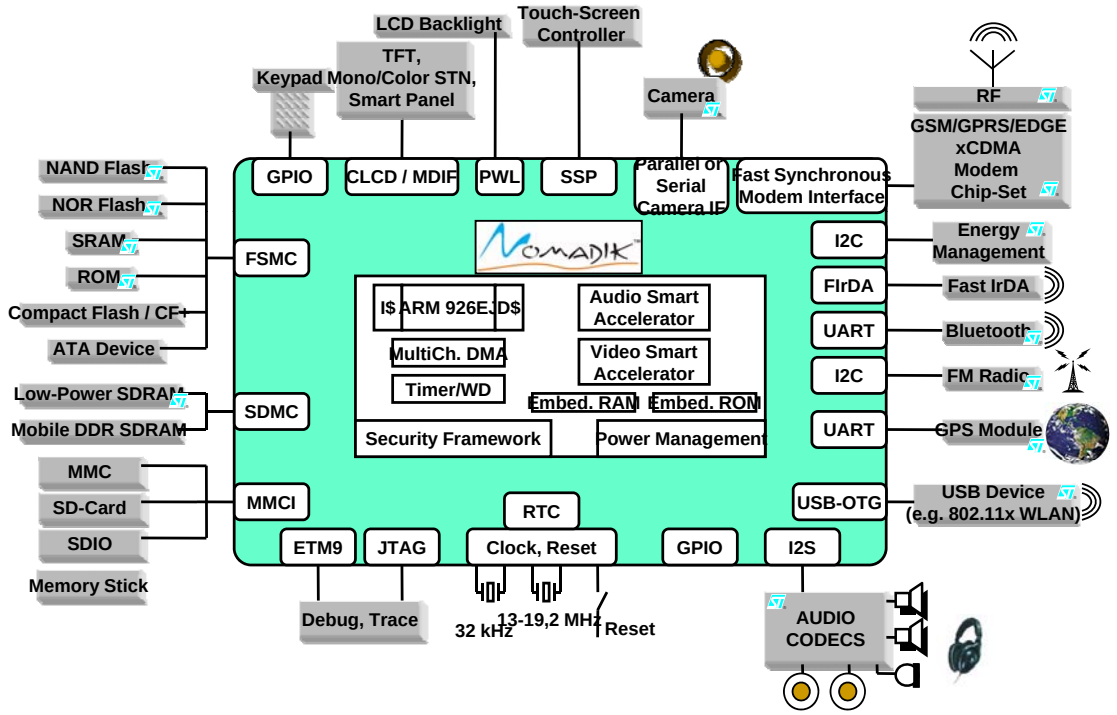
Two coprocessors
TCOP for turbo coding
VCOP for GSM AMR channels

Manufactured in 90 nm process technology



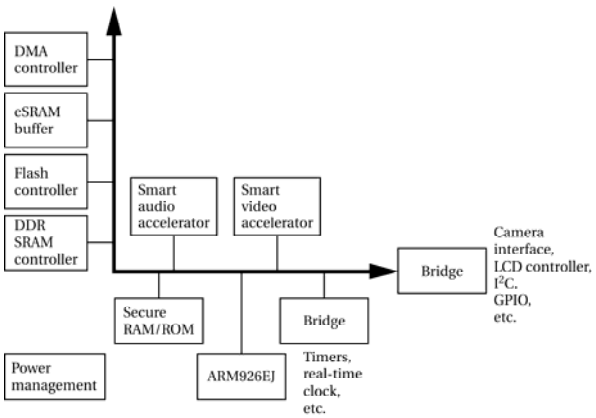
Note: The arrows show the direction from which the transfer originates.

4. Nomadik (ST)

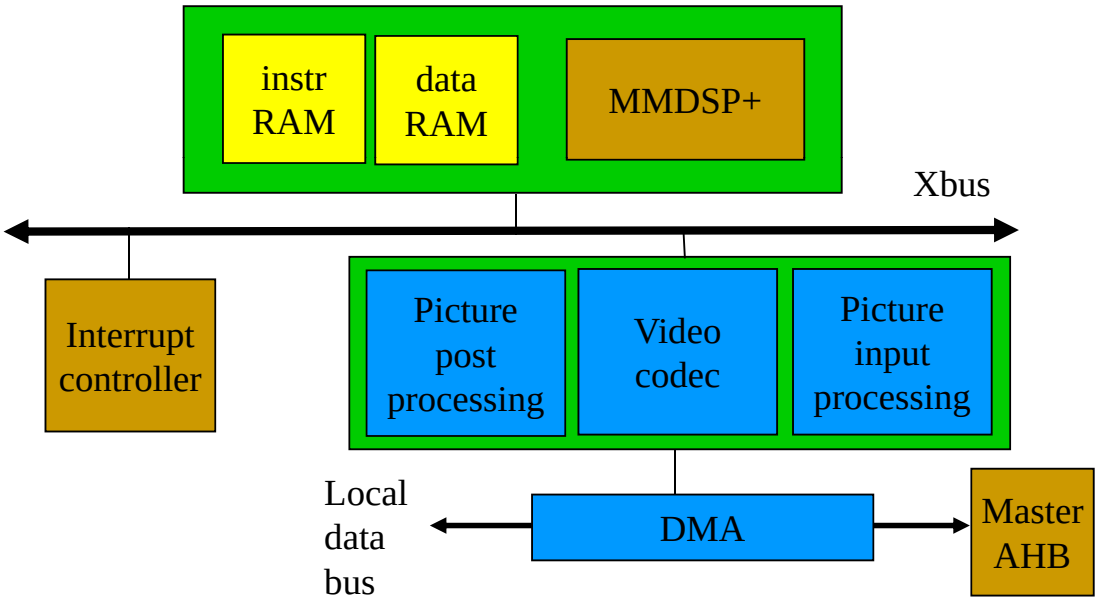


STMicro Nomadik

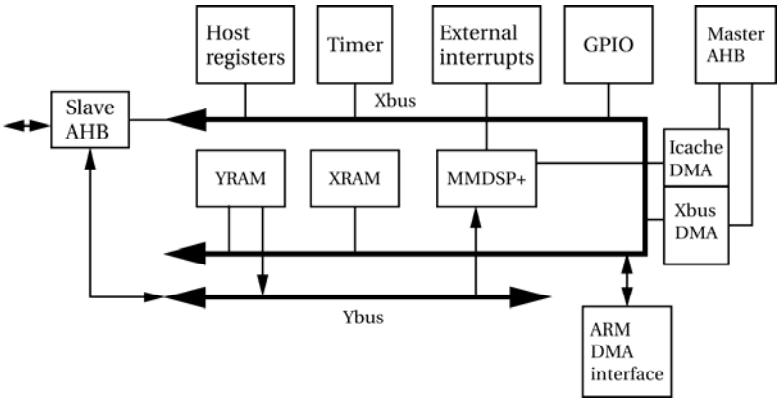
- Designed for mobile multimedia.
- Accelerators built around MMDSP+ core:
  - One instruction per cycle.
  - 16- and 24-bit fixed-point, 32-bit floating-point.



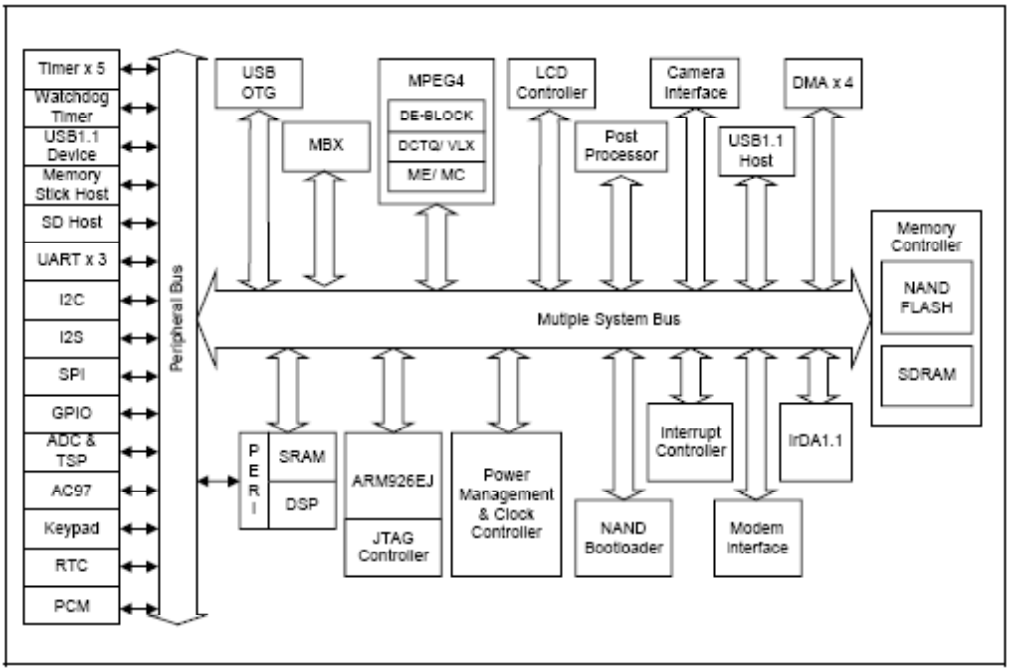
Nomadik video accelerator



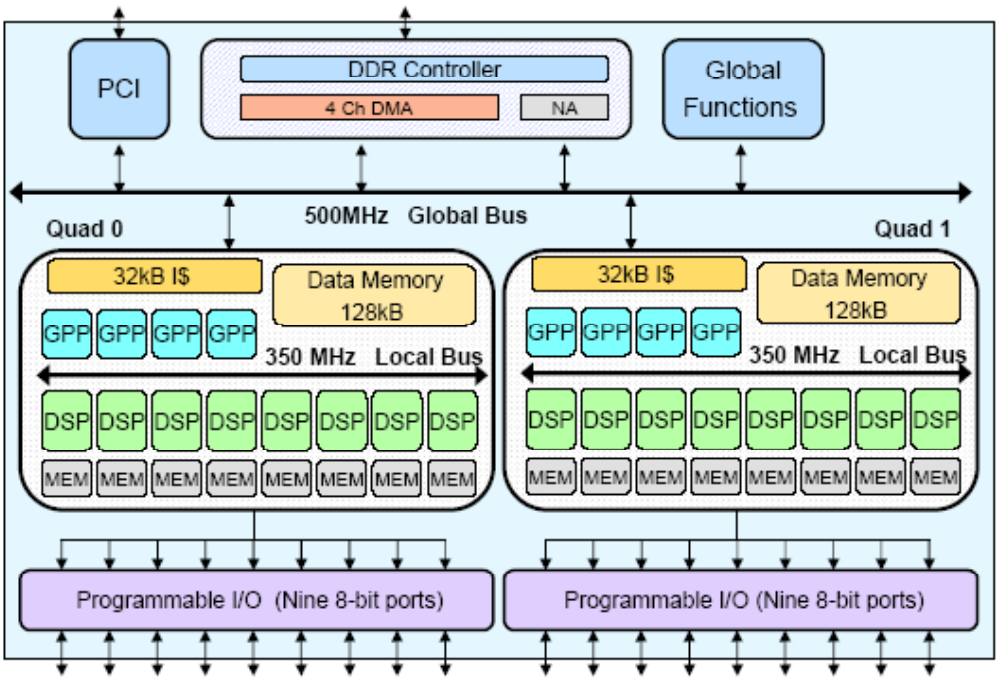
Nomadik audio accelerators



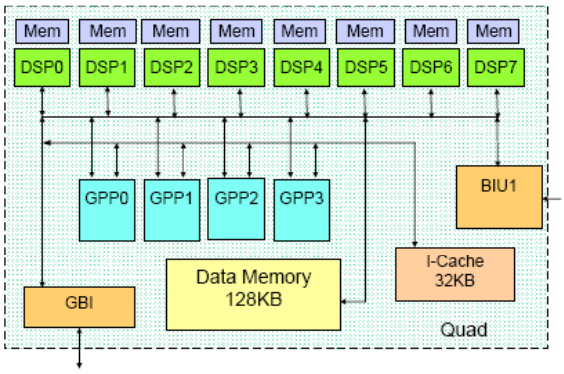
5. S3C2460 (Samsung)



6. CT3616 (Cradle)

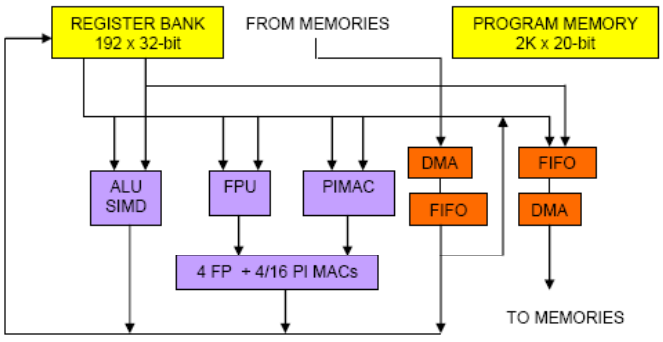


Quad in CT3616



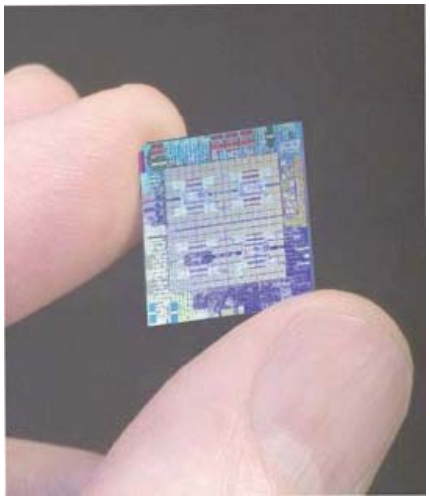
- 2 Compute subsystems with:
  - 8 DSP engines
  - 4 General Purpose Processors
  - 128KB of shared data memory
  - 32KB of shared instruction cache
  - Separate instruction & data buses
- Close I/O subsystem integration
  - Integrated bus interface unit
  - Efficient input stream pre-processing
  - Reduce DRAM accesses

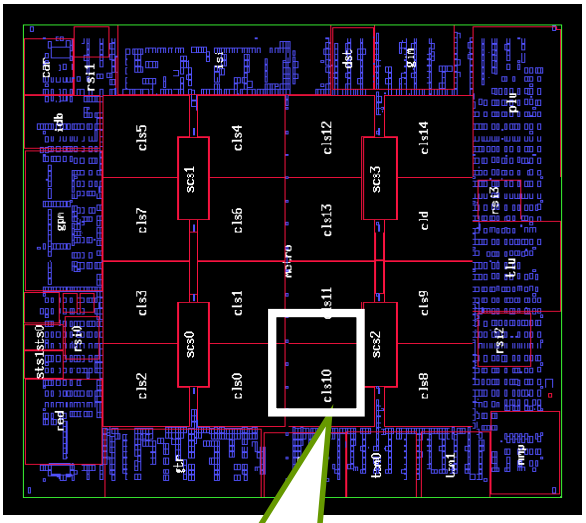
DSP core in CT3616



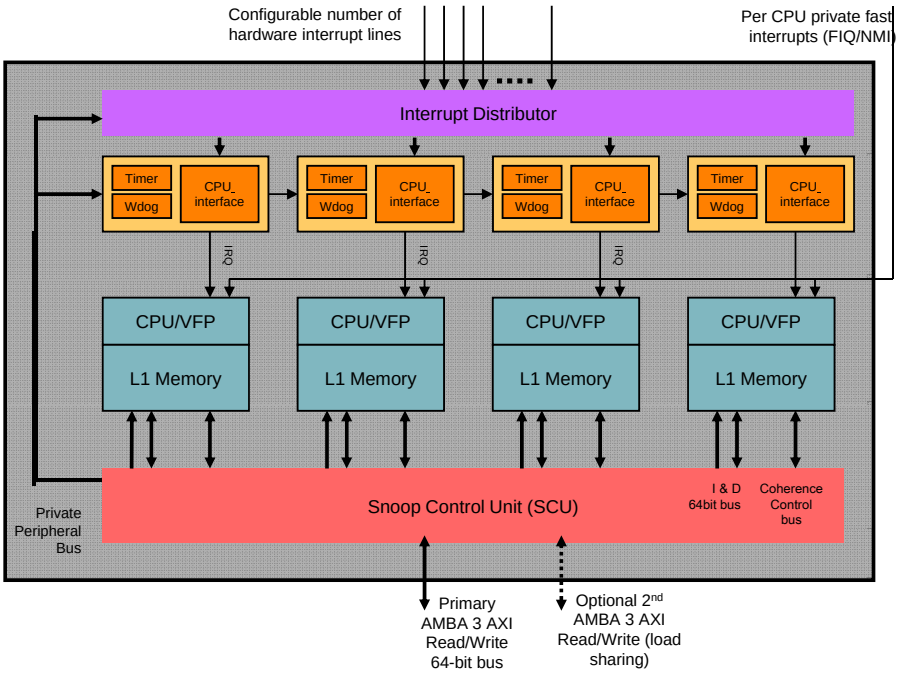
- 2 and 3 Operand instructions; operands can be packed
- High Memory Bandwidth provided by 192 registers with background transfers to/from local data memory
- MAC with 32 bit floating point and 4x8 bit, 2x16 bit, or 1x32 bit integer operands
- Floating point and packed integer MAC operations can be dispatched once per clock
- 16 parallel signed/unsigned 8x8 bit MAC operations per clock (256 MACs/cycle)
- SIMD operations on packed 8-bit or 16-bit data
- Single-cycle arbitrary bit-field access and packing

7. Silicon Packet Processor (Cisco)



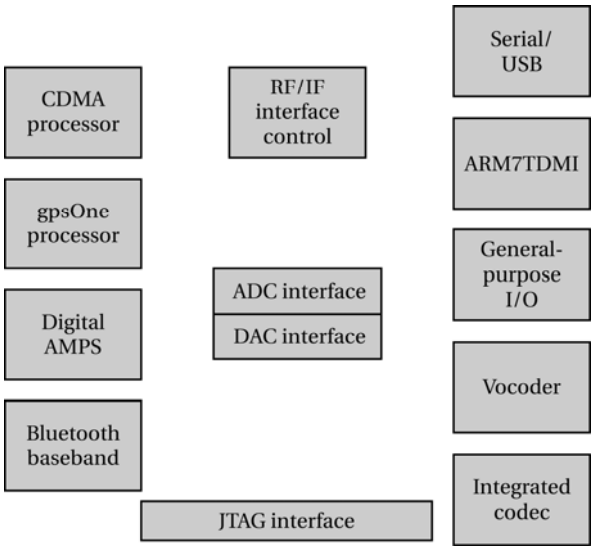


8. ARM11 MPCore

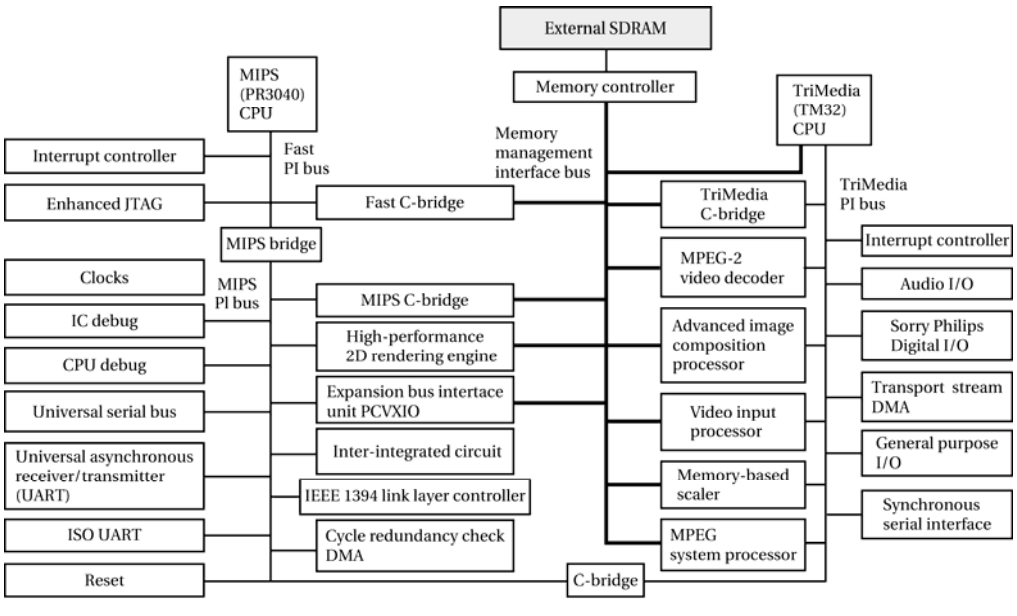


9. Qualcomm MSM5100

- 3G Cell phone system-on-chip.
- Two CDMA standards, analog cell phone standard.
- GPS, Bluetooth, music, mass storage.



10. Philips Viper Nexperia

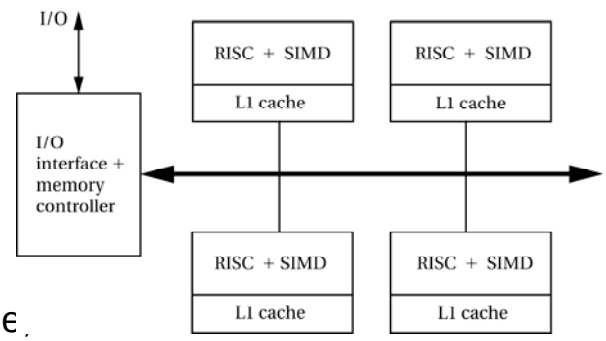


Viper Nexperia characteristics

- To build set-top boxes
- Designed to decode 1920 x 1080 HDTV.
- Trimedia TM32 VLIW processor runs video processing functions.
- MIPS PR3940 RISC CPU runs operating system.
- Synchronous DRAM interface for bulk storage.
- Variety of I/O devices.
- Accelerators: image composition, scaler, MPEG-2 decoder, video input processors, etc.

11. Lucent Daytona

- MIMD for signal processing.
- Processing element is based on SPARC V8 32-bit RISC processor.
- PE also includes reduced precision vector unit (RVU) has 16 x 64 vector register file with 5 ports.
- L1 cache of 8KB that is divided into 16 banks.
  - Configurable to Icache, Dcache, scratch pad memory
  - Set-associativity and size can be changed



Daytona split-transaction bus

- Every bus transaction carries a transaction ID that is used to match up the various part of the split transaction.
- A read is performed in two steps
- First, the address is sent in a four-cycle bus operation
  - Arbitrate for the address bus
  - Send the transaction data, including transaction ID, direction, address, size, and priority
  - Decode the transaction and determine the response
  - Respond with either retry, acknowledge, memory inhibit (for data modified cache), or shared.
- Second, the data is returned in a three-cycle bus operation.
  - Arbitrate for he data bus
  - Send the transaction ID
  - Send the data

Processing elements

- How many do we need?
- What types of processing elements do we need?
- Analyze performance/power requirements of each process in the application.
- Choose a processor type for each process.
- Determine which processes should share a processing element

Interconnection networks

- The interconnection network that connects the processing elements and memory is a key component of a complex multiprocessor.
- Its bandwidth is a key factor in performance.
- The network plays a critical role in power consumption.
- Client: sender or receiver connected to a network.
- Port: connection to a network on a client.
- Link: a connection between two clients.
  - half-duplex: only one direction at a time
  - full-duplex: can simultaneously transmit data in both directions.

Interconnection networks

- Network metrics:
  - Throughput: [Mbps] maximum allowable throughput. Concerned with the variations in data rates over time.
  - Latency: the amount of time it takes a packet to travel from one node and another. Best-case and worst-case latency.
  - Energy consumption: the amount of energy required to send a bit through the network.
  - Area: (1) silicon area of the transistors or (2) metal area of wires.
- Quality-of-service (QoS) is important for multimedia applications
  - Continuous or streaming data reliably and regularly
- On-chip or off-chip

Interconnection network models

- Three parts:
  - Source <-- line --> termination.
  - Lines may contain repeaters but no logic
  - The source and termination may contain registers but no multipacket buffers
- Throughput T, latency D.
- Link transmission energy E_b.
- Physical length L.

Traffic models

- Poisson model
  - When we are interested in the number of independent events that occur in a unit of time.
  - Traffic in telecommunication, phone calls
  - $P(X=n) = \mu^n e^{-\mu}/n!$ for n=0,1,…
  - $E(X) = \mu$, $Var(X) = \mu$.
- Poisson distribution
  - From binomial distribution
    - N -> ∞ and $Np = \mu$

Traffic models

- Streaming model
  - Data is produced periodically
  - Characterize data stream by the rate σ at which data are produced and the number of bits per datum (burstiness ρ).

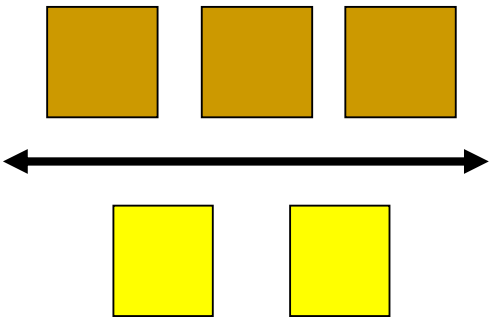
- **Intuitive Definition**
  - ***Burst*** *is a group of consecutive packets with shorter interpacket gaps than packets arriving before or after the burst of packets.*
  - ***Burstiness*** *is a characterization of bursts in a flow over T.*

Network topologies

- Major choices.
  - Bus.
  - Crossbar.
  - Buffered crossbar.
  - Mesh
  - Application-specific.

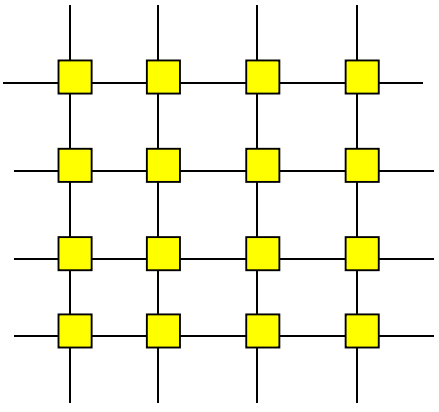
Bus network

- Throughput:
  - F: bus clock frequency, W: bus width
  - C: transaction overhead
  - $T \propto FW/(1+C)$.
- Advantages:
  - Well-understood.
  - Easy to program.
  - Many standards.
- Disadvantages:
  - Contention.
  - Significant capacitive load.



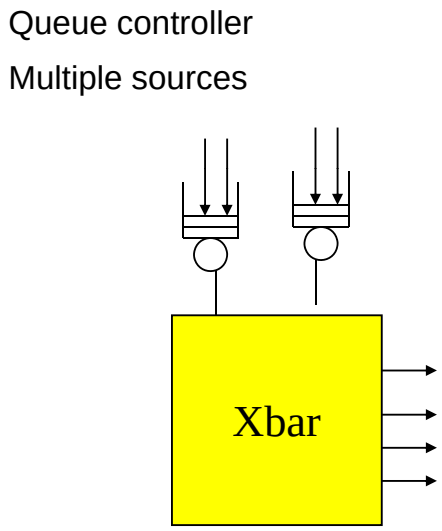
Crossbar

- Advantages:
  - No contention.
  - Broadcast, multicast
  - Simple design.
- Disadvantages:
  - Not feasible for large numbers of ports.
  - Complexity ~ $O(n^2)$



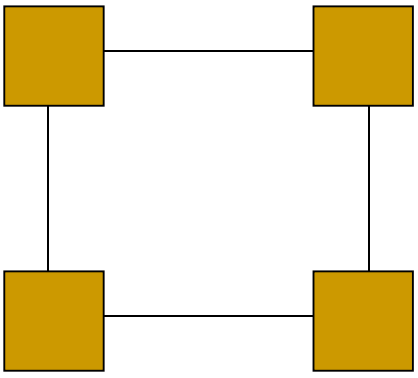
Buffered crossbar

- Advantages:
  - Smaller than crossbar.
  - Can achieve high utilization.
- Disadvantages:
  - Requires scheduling.



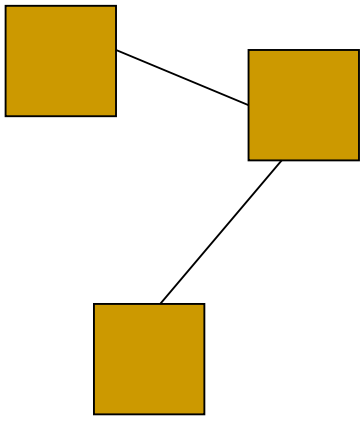
Mesh

- Full or partial mesh.
- Advantages:
  - Well-understood.
  - Regular architecture.
- Disadvantages:
  - Poor utilization.



Application-specific.

- Advantages:
  - Higher utilization.
  - Lower power.
- Disadvantages:
  - Must be designed.
  - Must carefully allocate data.



Layers in the operation of the interconnection network

- Physical layer
  - refers to link-level protocols for transferring messages and otherwise managing the physical channels between adjacent router
- Switching layer
  - utilizes these physical layer protocols to implement mechanisms for forwarding messages through the network.
- Routing layer
  - makes routing decisions to determine candidate output channels at intermediate router nodes and thereby establish the path through the network.

Router model

- Buffers: FIFO buffers
- Switch: crossbar switches for full connectivity
- Routing and arbitration unit
  - Implement the routing algorithm
  - Select the output link for an incoming message and accordingly set the switch
- Link controller (LC)
  - Control the flow of message across the physical channel
- Processor interface
  - Injection channels : from the processor
  - Ejection channels: to the processor

Router model

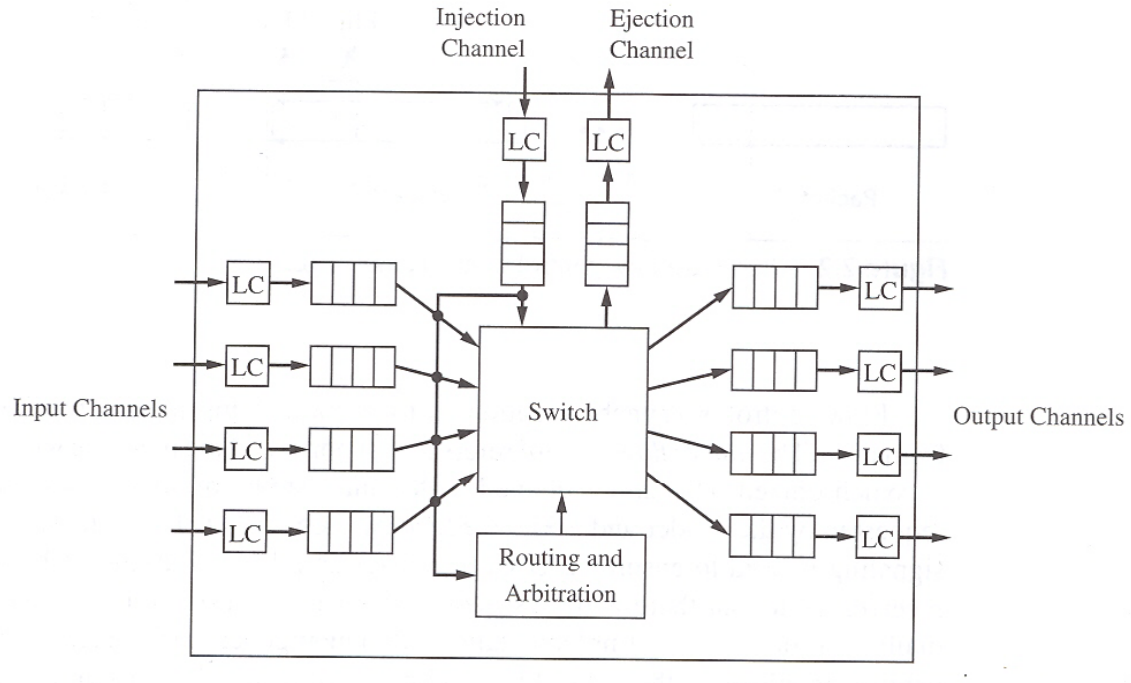


Figure 2.1 Generic router model. (LC = link controller.)

Flow control units in a message

- A message may be partitioned into fixed-length of packets
- Flits: a message flow control unit
  - Packets may be broken into flits
- Phits: a physical flow control unit
  - Due to channel width constraints, multiple physical channel cycles may be used to transfer a single flit.

Router model

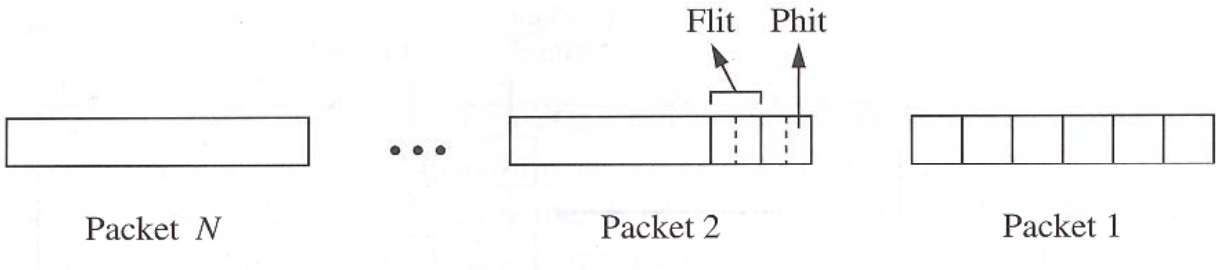


Figure 2.2 Alternative flow control units in a message.

Circuit switching

- A physical path form the source to the destination is reserved prior to the transmission of the data
- This is realized by injecting the routing header flit into the network.
- This routing probe contains the destination address and some additional control information.
- When the probe reaches the destination, a complete path has been set up and an acknowledgement is transmitted back to the source.
- Then the message content may now be transmitted at the full bandwidth of the hardware path.

Router model

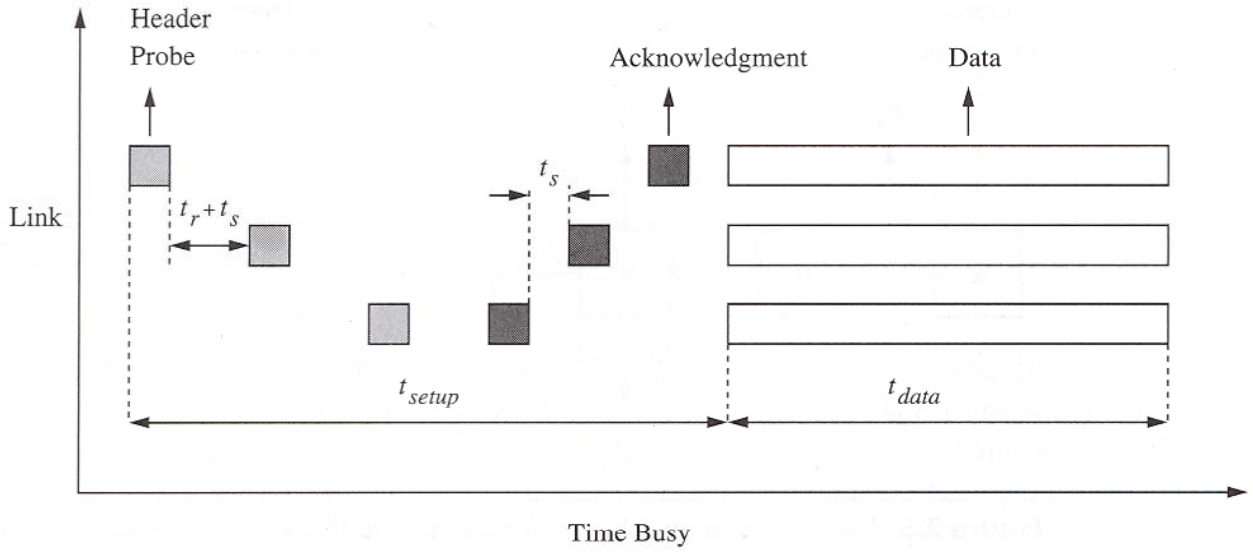


Figure 2.6 Time-space diagram of a circuit-switched message.

Packet switching

- The message can be partitioned and transmitted as fixed-length packets
- Packet header: contain routing and control information
- Each packet is individually routed from source to destination.
- A packet is completely buffered at each intermediate node before it is forwarded to the next node.
- that is why it is also referred to store-and-forward (SAF) switching.

Packet switching

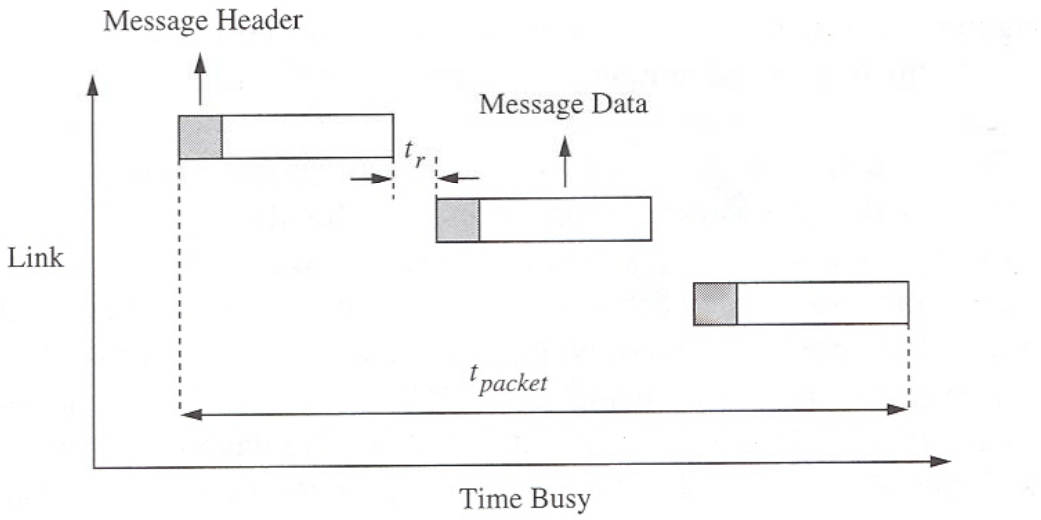


Figure 2.8 Time-space diagram of a packet-switched message.

Virtual cut-through (VCT) switching

- Virtual cut-through ensures that the entire path is available before starting transmission.
- Rather than waiting for the entire packet to be received, the packet header can be examined as soon as it is received.
- As soon as routing decision have been made, the current router can start forwarding the header and following data bytes and can cut through to the input of the next router.
- If the header is blocked on a busy output channel, the complete message is buffered at the node. Thus, at high network loads, VCT switching behaves like packet switching.

VCT switching

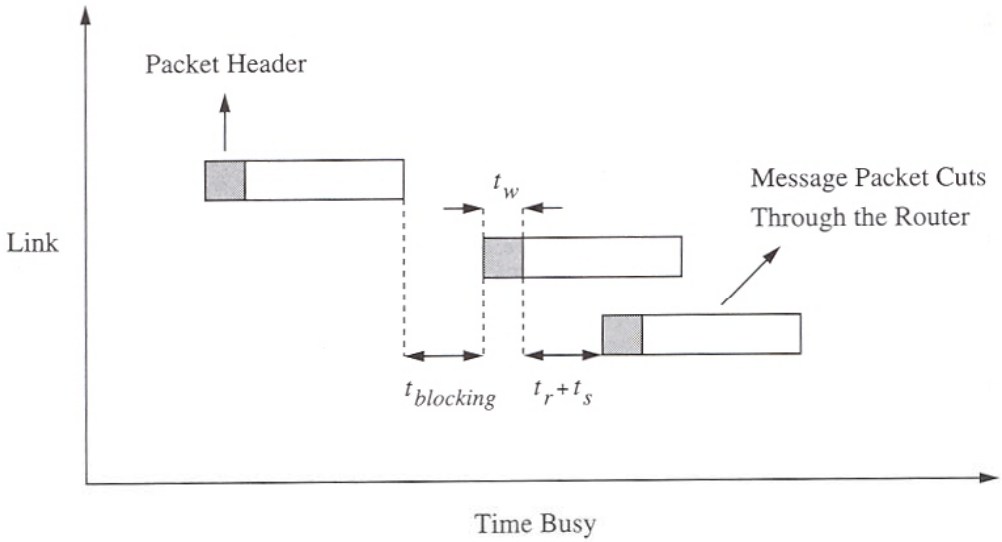


Figure 2.10 Time-space diagram of a virtual cut-through switched message. ($t_{blocking}$ = Waiting time for a free output link.)

Wormhole switching

- Message packets are also pipelined through the network. However, the buffer requirement within the routers are substantially reduced over the requirements for VCT switching.
- A message packet is broken into flits. The flits is the unit of message flow control.
- Input and output buffers at a router are typically large enough to store a few flits.

Wormhole switching

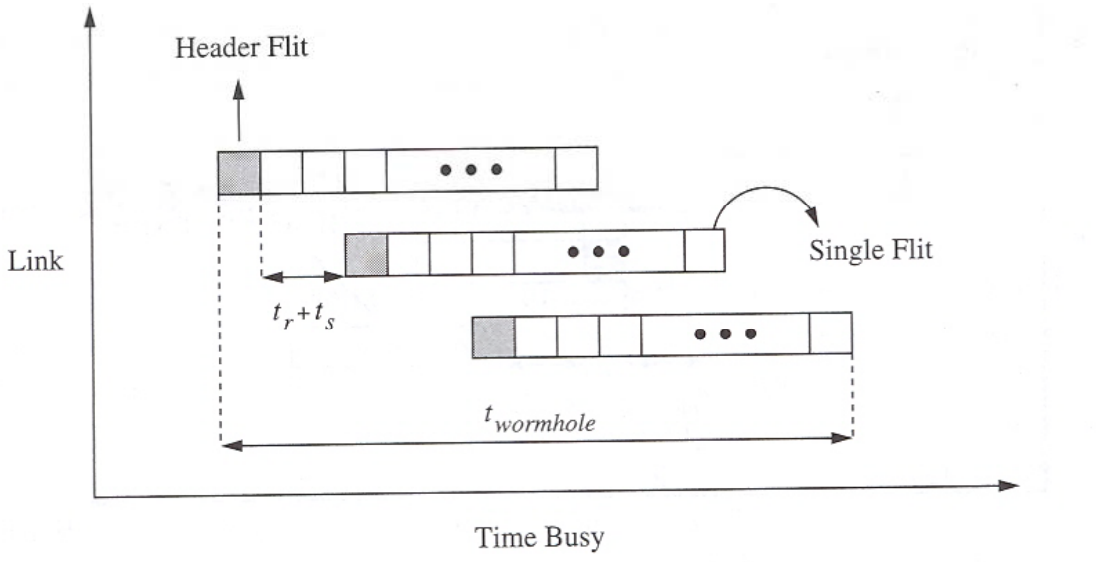


Figure 2.11 Time-space diagram of a wormhole-switched message.

Wormhole switching

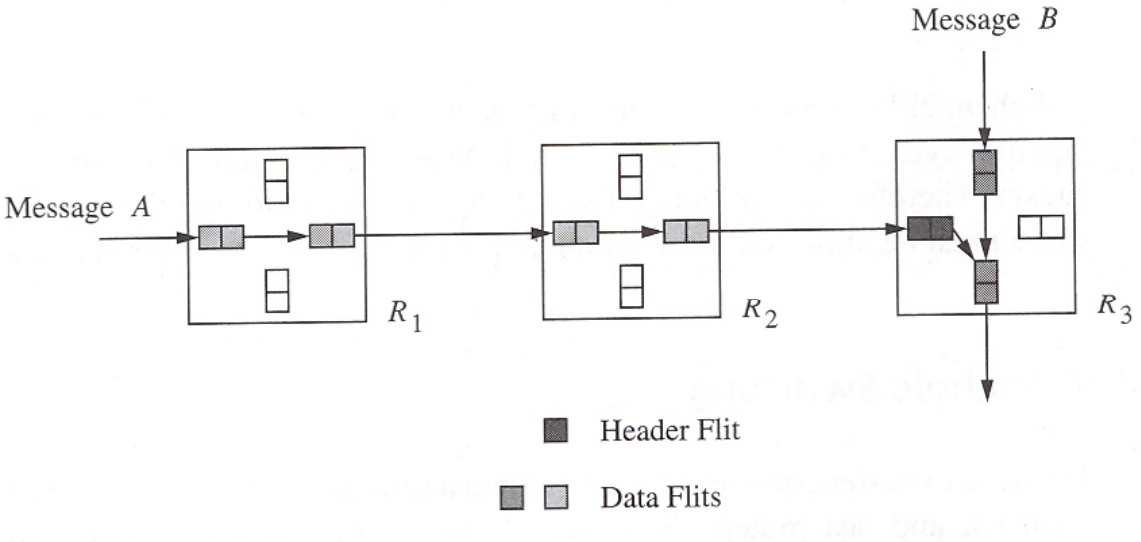


Figure 2.12 An example of a blocked wormhole-switched message.

Virtual Channels

- The message can be partitioned and transmitted as fixed-length packets.
- Once a message occupies a buffer for a channel, no other message can access the physical channel, even if the message is blocked.
- Alternatively, a physical channel may support several logical or virtual channels multiplexed across the physical channel.
- Virtual channel were originally introduced to solve the problem of deadlock in wormhole-switched networks.

Virtual channel

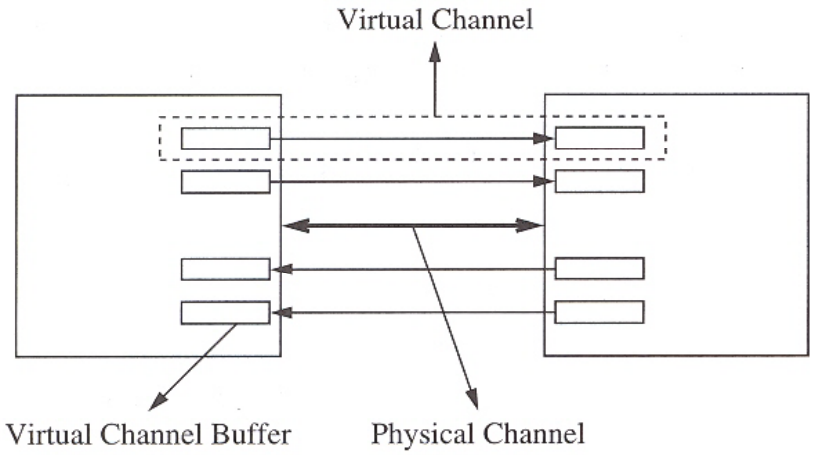


Figure 2.17 Virtual channels.

Virtual channel

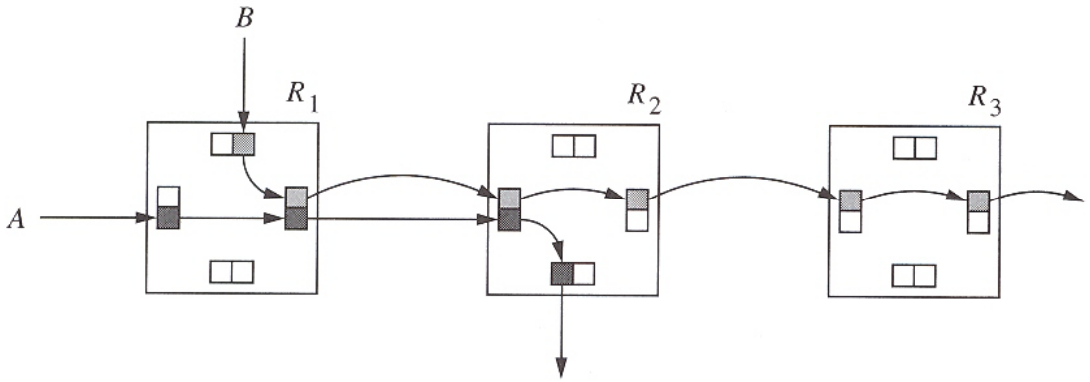


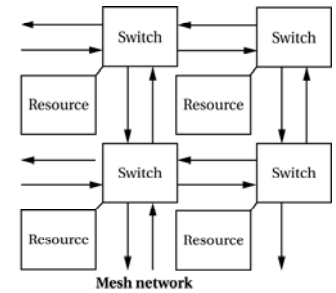
Figure 2.18 An example of the reduction in header blocking delay by using two virtual channels for each physical channel.

Networks-on-chips

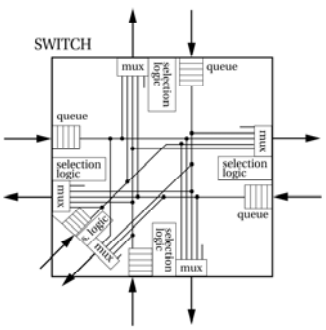
- Help determine characteristics of MPSoC:
  - Energy per operation.
  - Performance.
  - Cost.
- NoCs do not have to interoperate with other networks.
  - NoCs have to connect to existing IP, which may influence interoperability.
- QoS is an important design goal.

Nostrum network

- A mesh network---switch connects to four nearest neighbors and a local processor or memory.
- Each switch has queue at each input.
- Selection logic determines order in which packets are sent to output links.



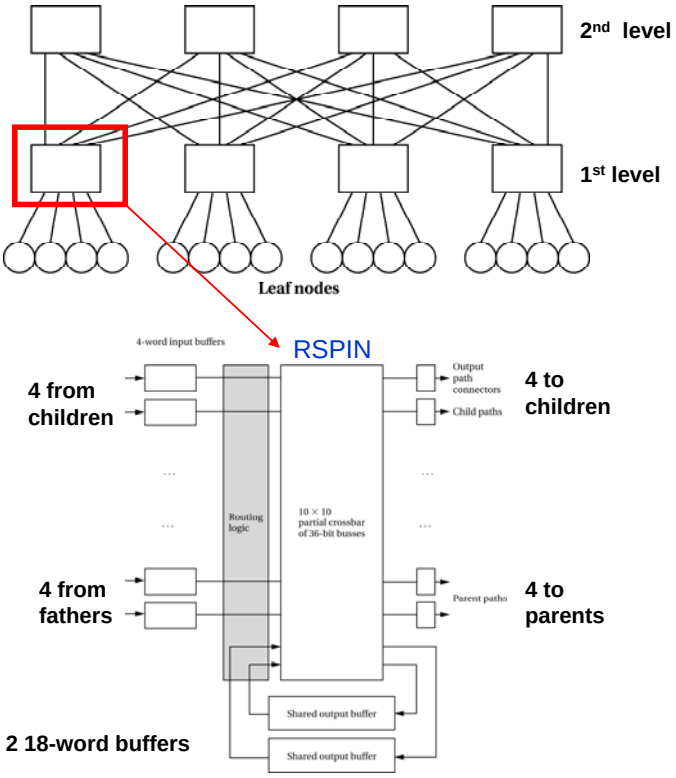
Mesh network



Switch block diagram

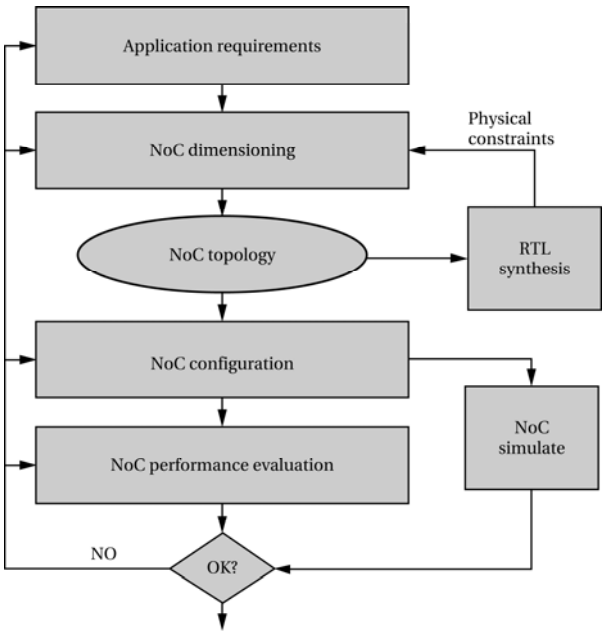
SPIN network

- Scalable programmable interconnection network
- A scalable network based on a fat-tree.
  - Bandwidth of links gets larger toward root of tree.
- The network is designed to rout messages in a treelike style, moving up the tree and then back down.
- All routing nodes use the same routing function.
- Leaf nods are the processors and memory elements
- The SPIN network used two 32-bit data paths: one for each direction, to provide full-duplex links.



Goossens et al. NoC methodology

- Intended to design networks for QoS-intense applications such as multimedia.
- Based on several principles
  - Uses QoS to characterize resource usage.
  - Uses calibration to estimate the characteristics of unpredictable resources.
  - Uses resource management to allocate shared resources.
  - Uses guaranteed services to allocate resource to users.

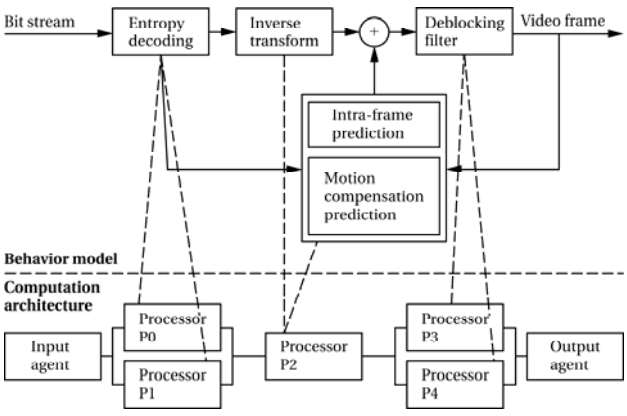


Coppola et al. OCCN methodology

- A methodology and tool set for on-chip communication systems.
- OCCN models iter-module communication using three layers:
  - NoC communication layer: implements lower layers of OSI stack.
  - Adaptation layer: uses hardware and software to implement OSI middle layers.
  - Application layer: built on top of communication API.

Xu et al. H.264 network design

- Designed NoC for H.264 decoder.
- Process -> PE mapping was given.
- Compared RAW mesh, application-specific networks.



Computation architecture

Xu et al. H.264 network design

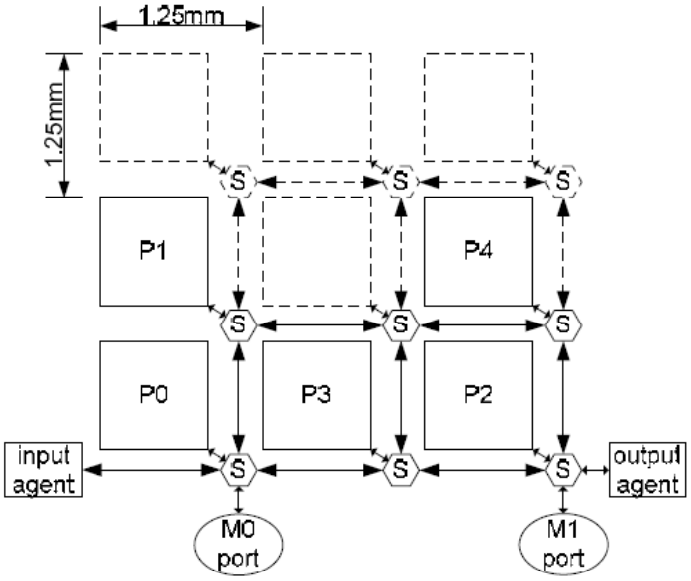
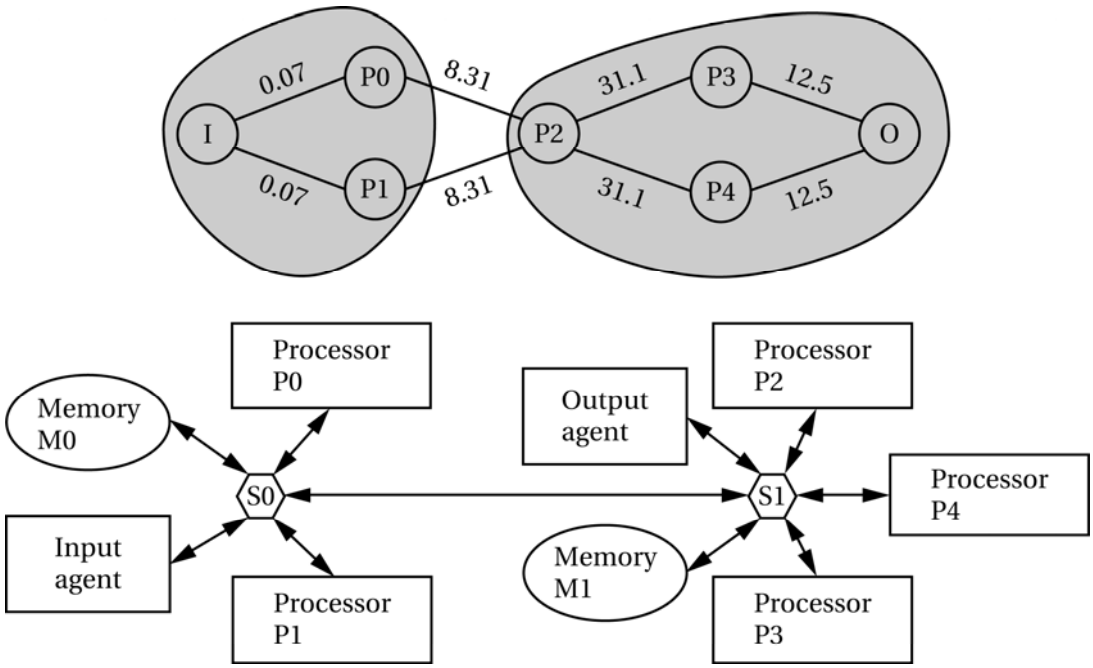


Figure 3. RAW architecture

Application-specific network for H.264



RAW/application-specific network comparison