Computer System Architecture

MODULE II

Elizabeth Isaac
Dept of Computer Science &Engg
M A College Kothamangalam

Processors And Memory Hierarchy

Advanced Processor Technology

- Design Space of Processors
- Instruction-Set Architectures
- CISC Scalar Processors
- RISC Scalar Processors

Superscalar and Vector Processors

- Superscalar Processors
- The VLIW Architecture
- Vector and Symbolic Processors

Memory Hierarchy Technology

- Hierarchical Memory Technology
- Inclusion, Coherence, and Locality
- Memory Capacity Planning

Advanced Processor Technology

- CISC-Complex-Instruction-Set Computing
- RISC-Reduced-Instruction-Set Computing
- superscalar
- VLIW- Very long Instruction Word
- superpipelined vector
- symbolic processors.

Design Space Of Processors

- clock rate versus cycles per instruction(CPI).

Continue…

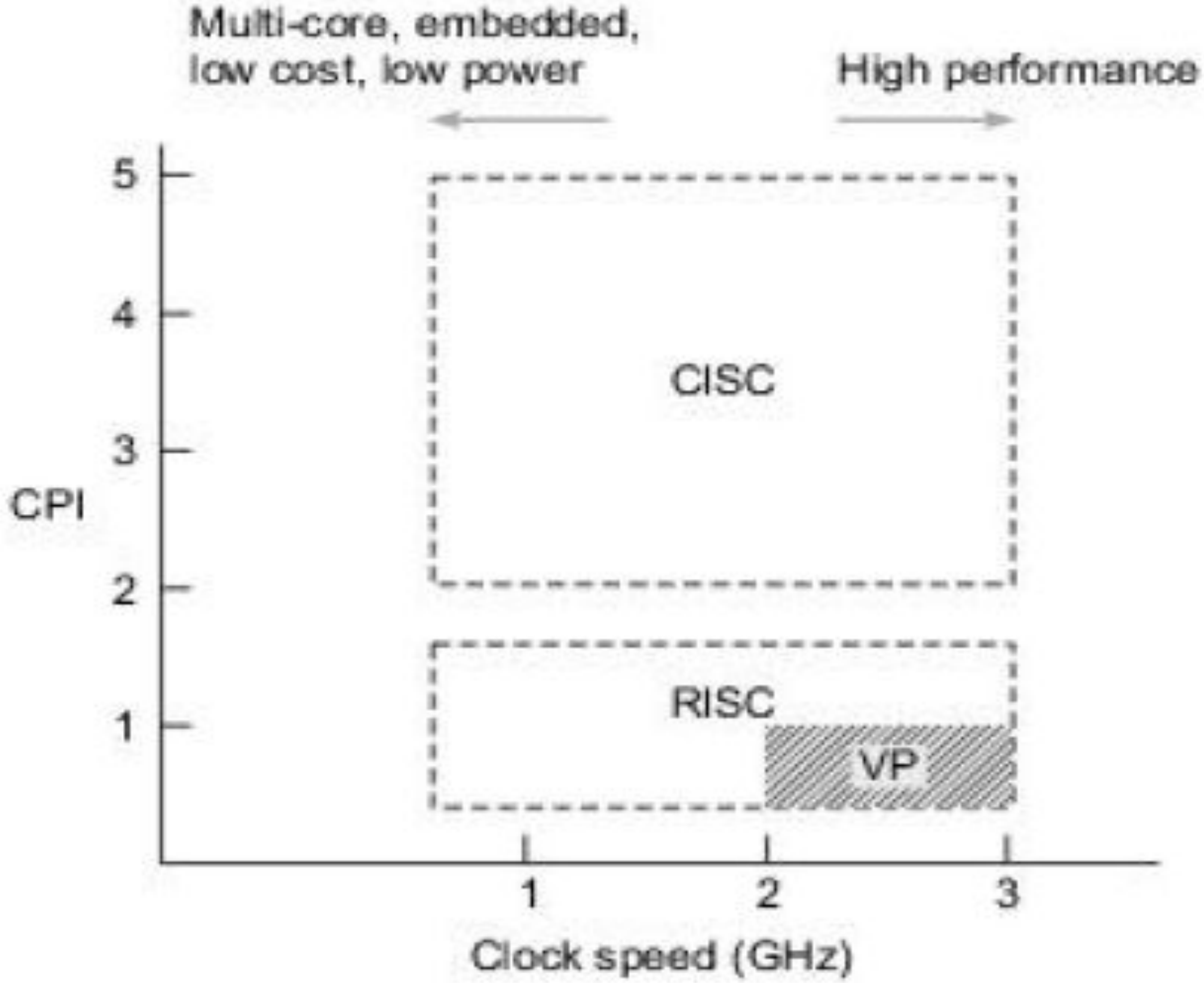
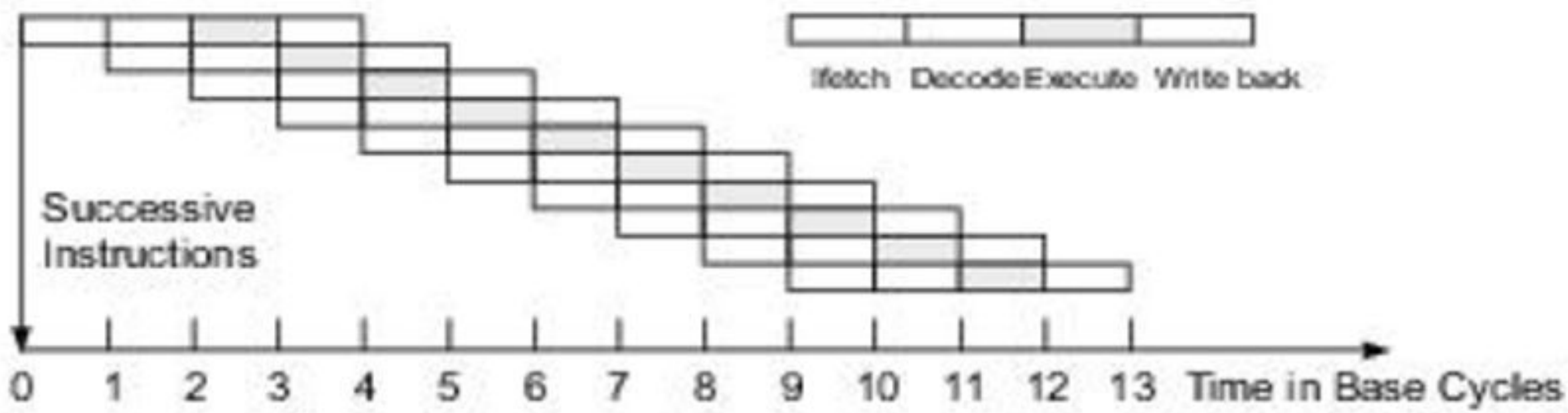


Fig. 4.1 CPI versus processor clock speed of major categories of processors

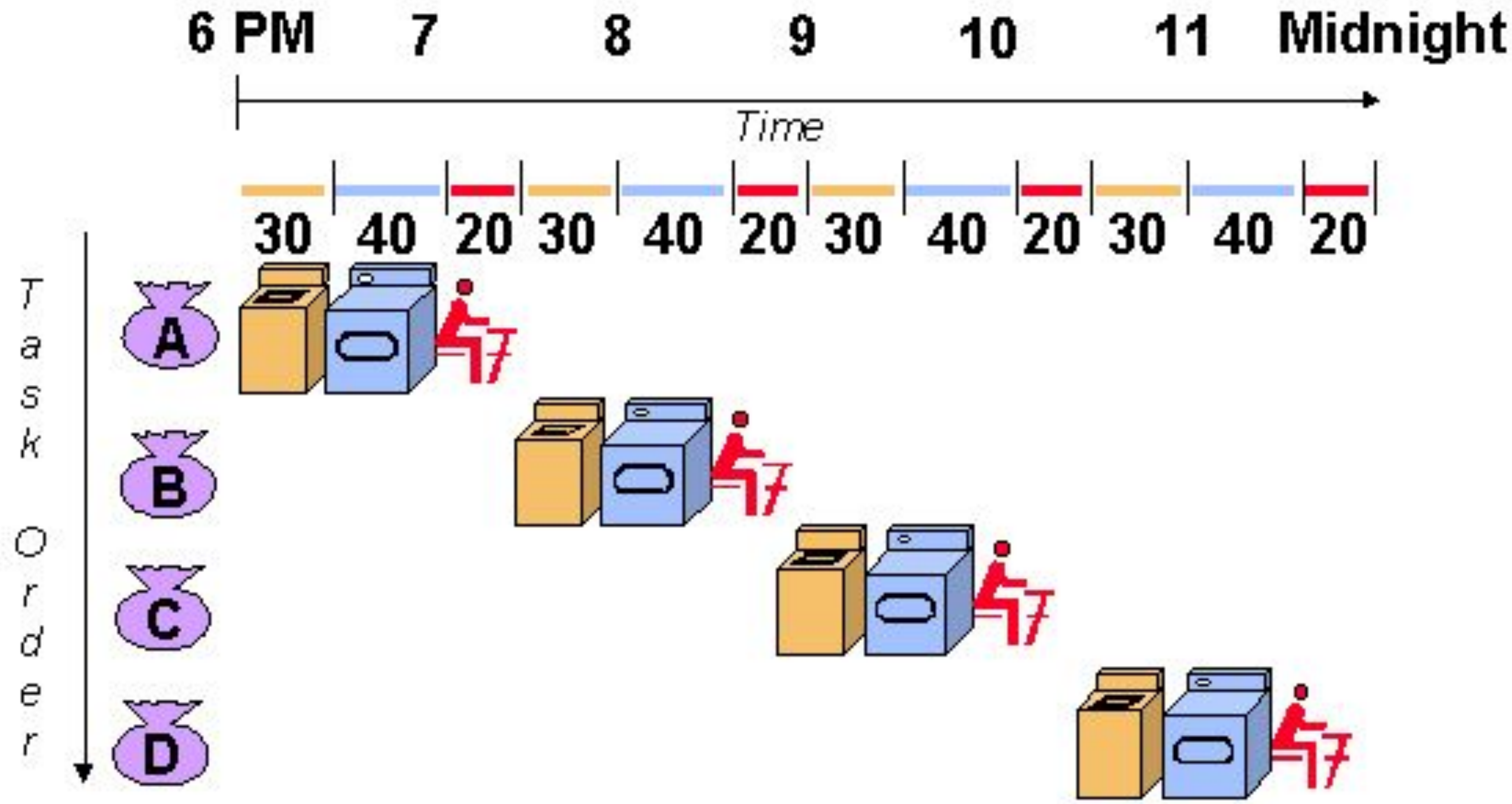
Instruction Pipeline

- Fetch
- Decode
- Execute
- Write-back

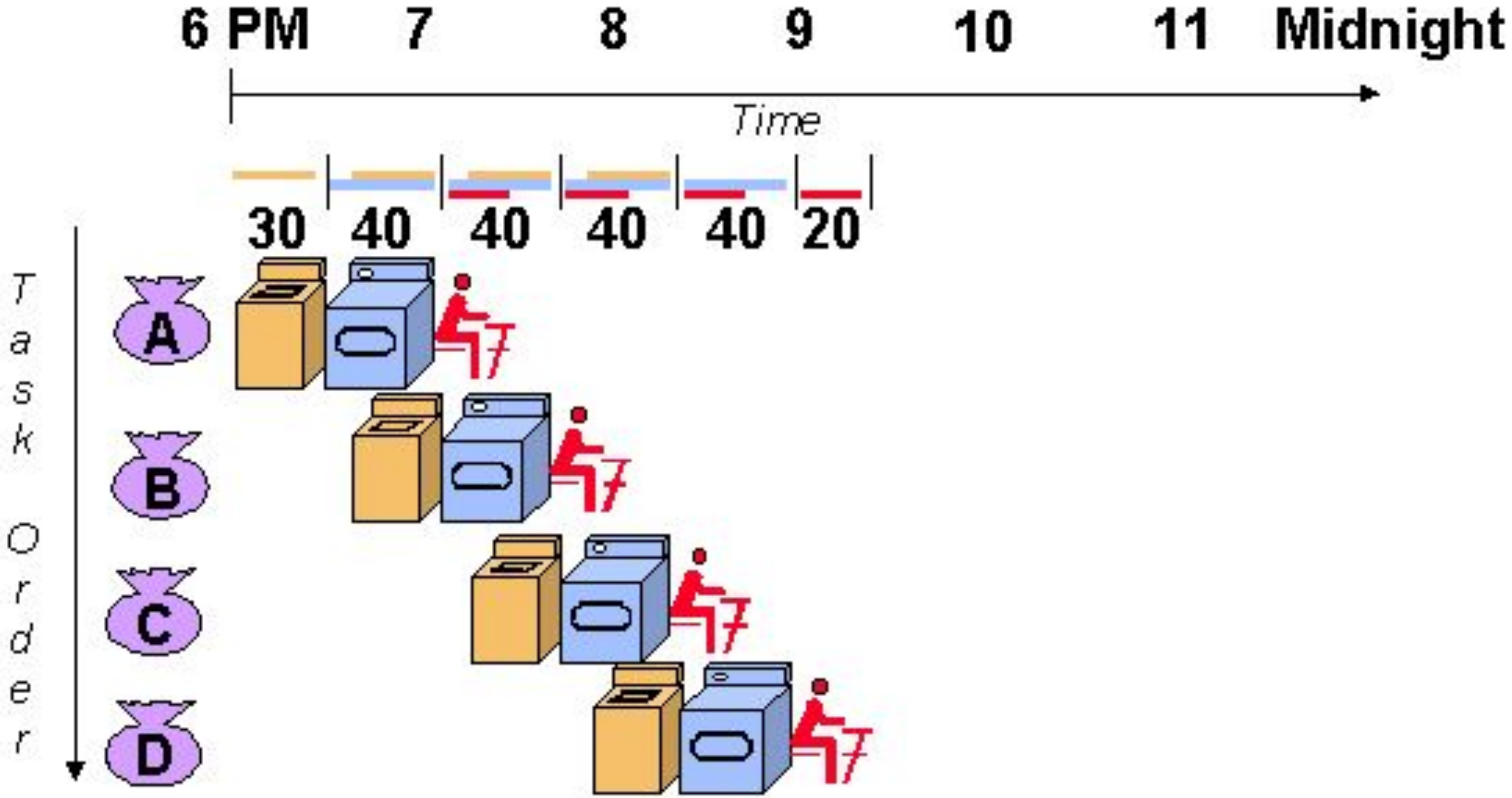


(a) Execution in a base scalar processor

Example


6 PM
7
8
9
10
11
Midnight
Time
30
40
20
30
40
20
30
40
20
30
40
20
Task Order
A
B
C
D

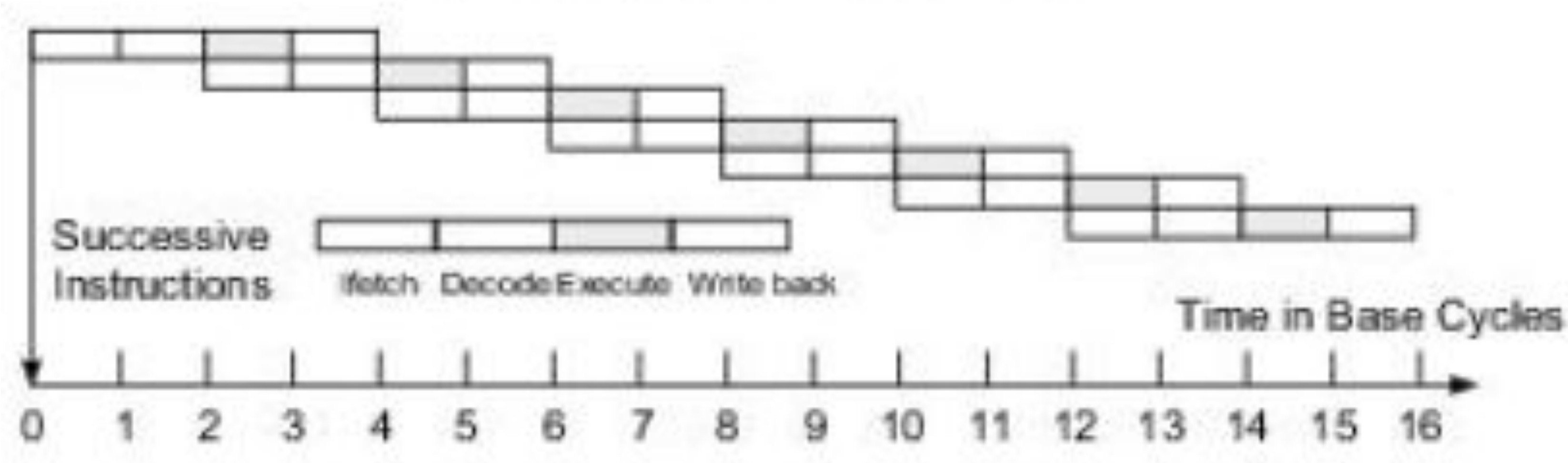
example



Instruction Pipeline Cont…

- Instruction pipeline cycle - the clock period of the instruction pipeline.
- Instruction issue latency - the time (in cycles) required between the issuing of two adjacent instructions.

Instruction Pipeline

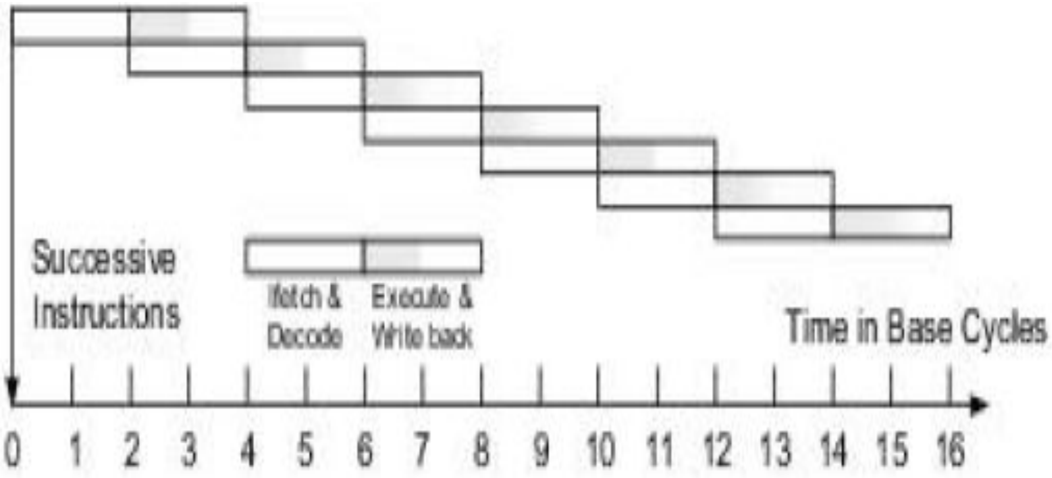


(b) Underpipelined with two cycles per instruction issue

Instruction Pipeline

- Instruction issue rate - the number of instructions issued per cycle, also called the degree of a superscalar processor.
- Simple operation latency
- Resource conflicts - two or more instructions demand same functional unit at the same time.

Instruction Pipeline Cont…



(c) Underpipelined with twice the base cycle

Fig. 4.2 Pipelined execution of successive instructions in a base scalar processor and in two underpipelined cases (Courtesy of Jouppi and Wall; reprinted from *Proc. ASPLOS*, ACM Press, 1989)

Instruction Pipeline Cont…

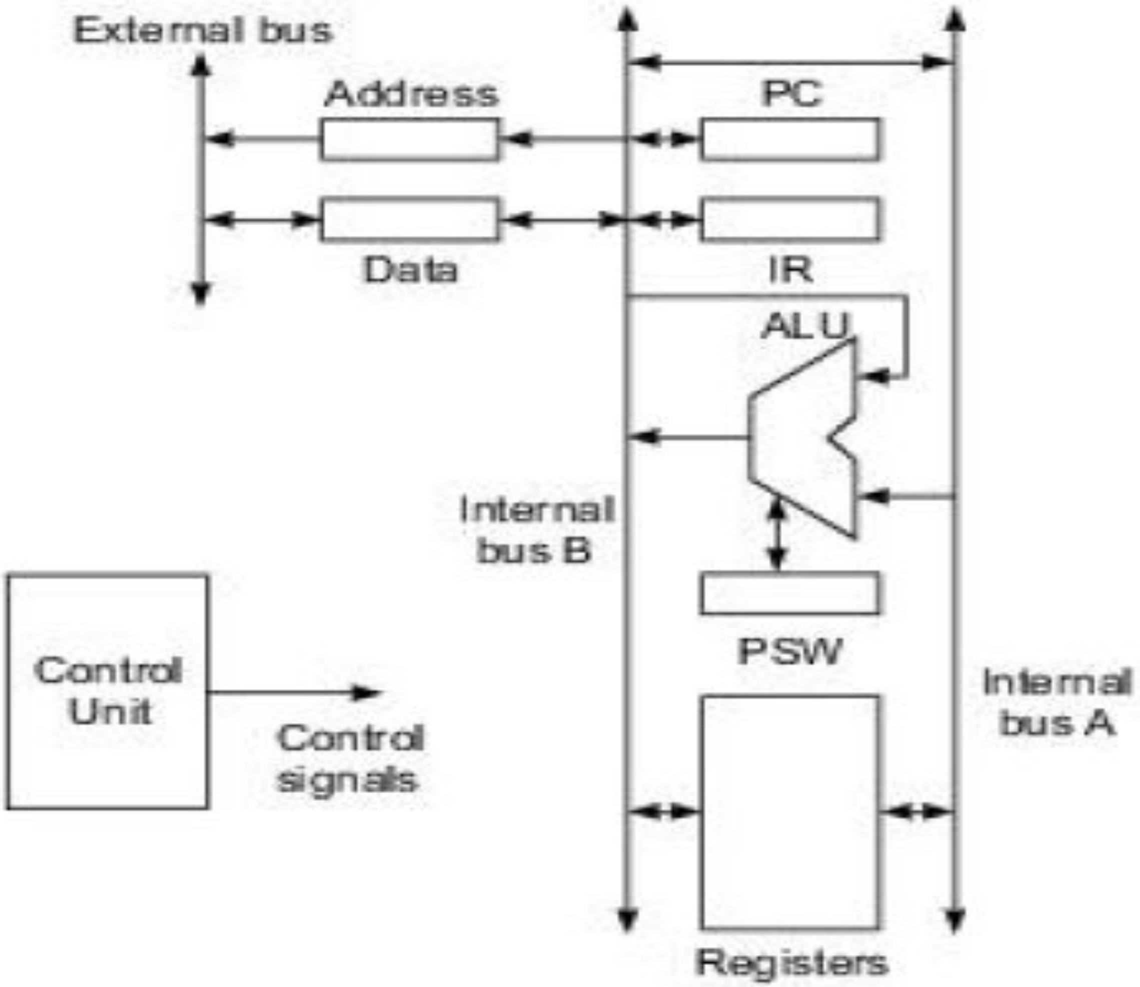


Fig. 4.3 Data path architecture and control unit of a scalar processor

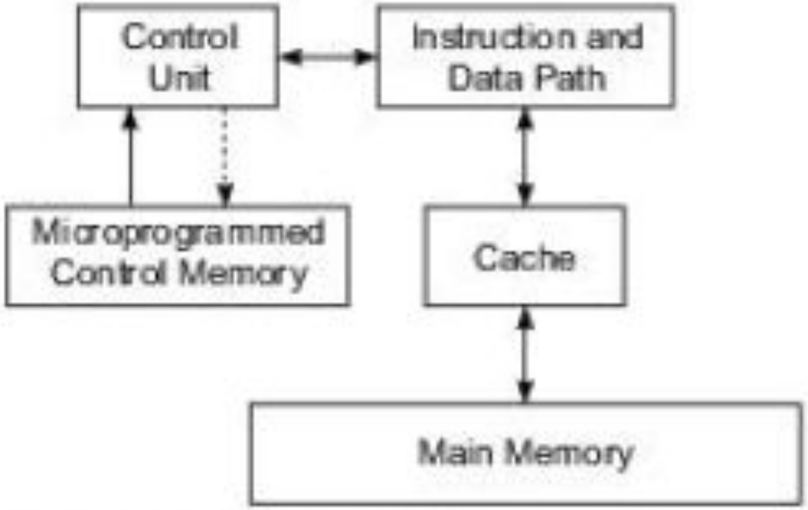
Instruction Set Architectures

- CISC and RISC.

- RISC
  - Execution time is very less
  - Decoding of instructions is simple
  - processors are highly pipelined
  - The most common RISC microprocessors are Alpha, ARC, ARM, AVR, MIPS, PA-RISC, PIC, Power Architecture, and SPARC.

- CISC
  - Execution time is very high.
  - Decoding of instructions is complex
  - They are normally not pipelined or less pipelined
  - Examples of CISC processors are the System/360, VAX, PDP-11, Motorola 68000 family, AMD and Intel x86 CPUs

Instruction Set Architectures Cont…



(a) The CISC architecture with microprogrammed control and unified cache

(b) The RISC architecture with hardwired control and split instruction cache and data cache

Distinctions between typical RISC and typical CISC processor architectures (Courtesy of Gordon Bell, 1989)

Instruction Set Architectures Cont…

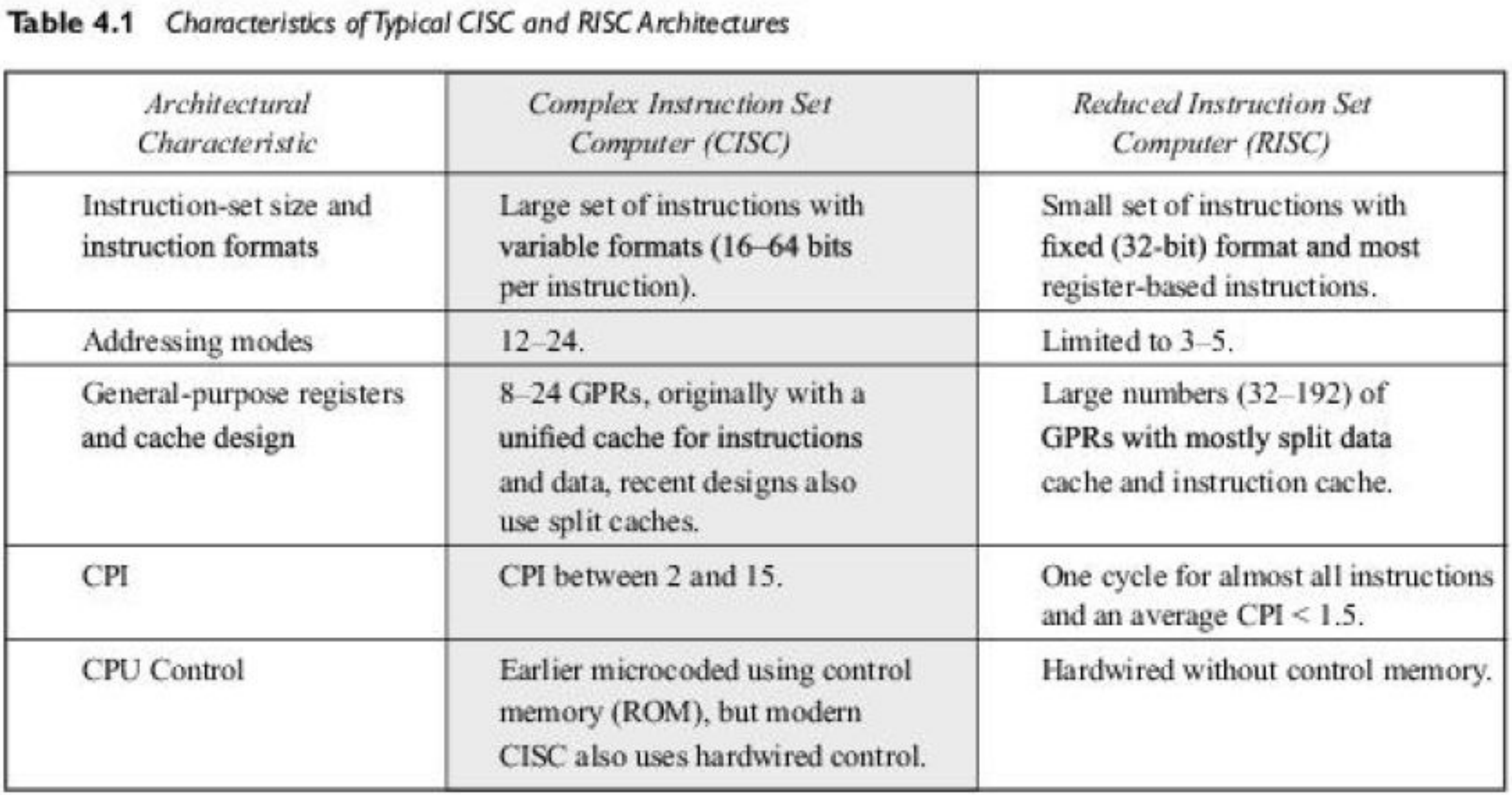


Table 4.1 *Characteristics of Typical CISC and RISC Architectures*

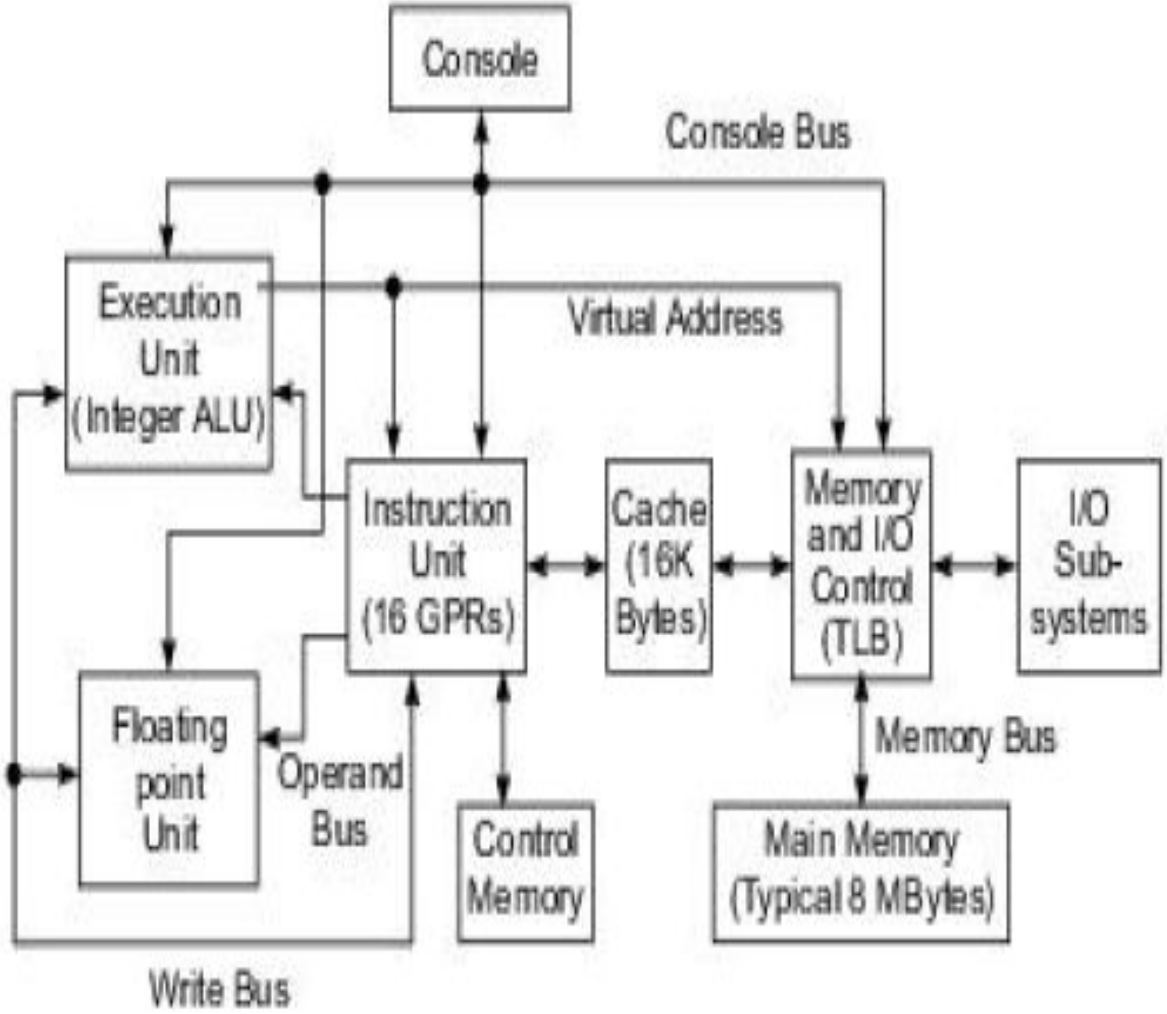
Architectural Characteristic	*Complex Instruction Set Computer (CISC)*	*Reduced Instruction Set Computer (RISC)*
Instruction-set size and instruction formats	Large set of instructions with variable formats (16–64 bits per instruction).	Small set of instructions with fixed (32-bit) format and most register-based instructions.
Addressing modes	12–24.	Limited to 3–5.
General-purpose registers and cache design	8–24 GPRs, originally with a unified cache for instructions and data, recent designs also use split caches.	Large numbers (32–192) of GPRs with mostly split data cache and instruction cache.
CPI	CPI between 2 and 15.	One cycle for almost all instructions and an average CPI < 1.5.
CPU Control	Earlier microcoded using control memory (ROM), but modern CISC also uses hardwired control.	Hardwired without control memory.

CISC Scalar Processors Cont…

The Digital Equipment VAX 8600 Processor Architecture

The Motorola MC68040 microprocessor

VAX 8600 Processor



Captions:
CPU = Central Processor Unit
TLB = Translation Lookaside Buffer
GPR = General Purpose Register

Fig. 4.5 The VAX 8600 CPU, a typical CISC processor architecture (Courtesy of Digital Equipment Corporation, 1985)

VAX 8600 Processor

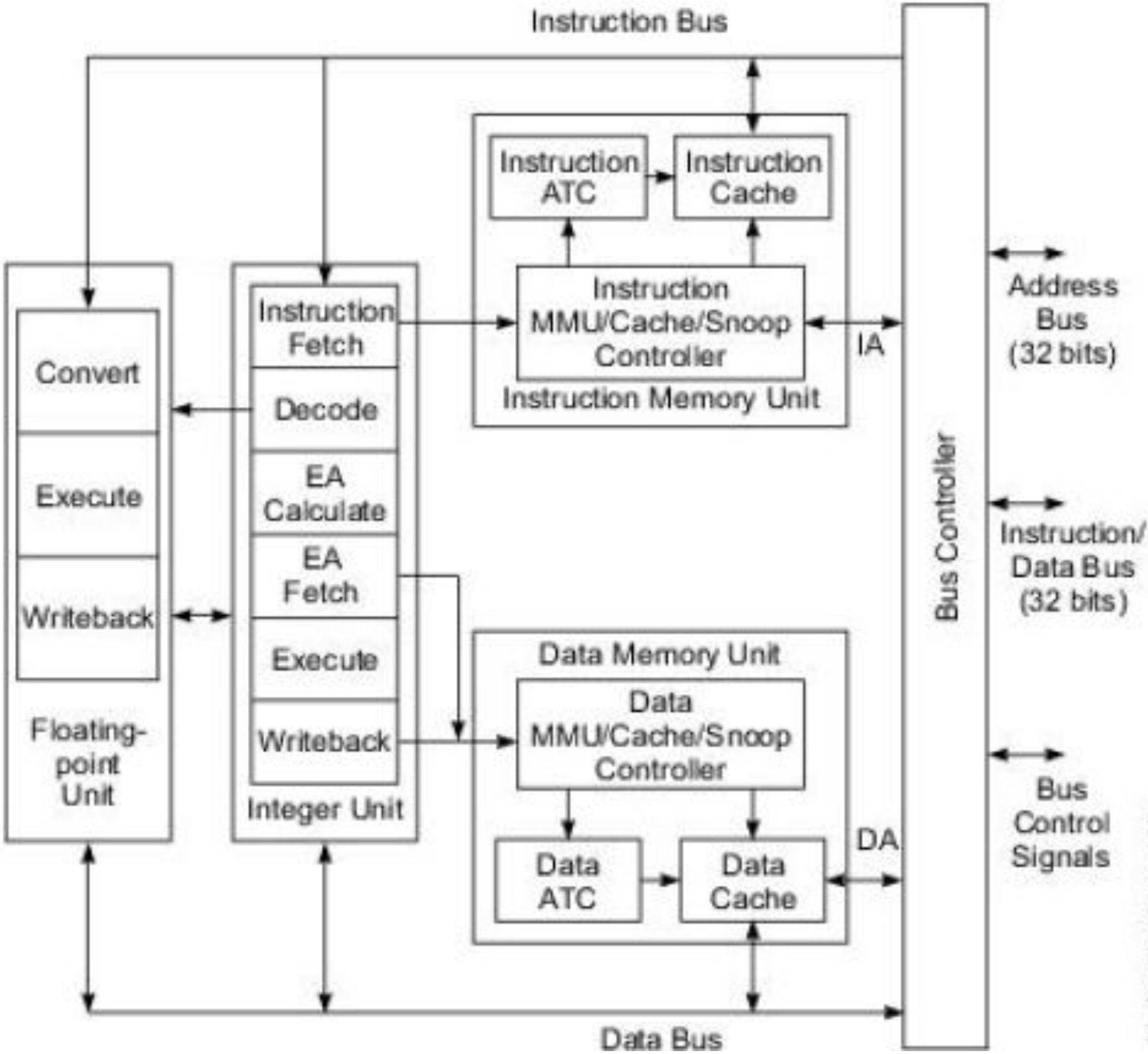
VAX 8600 processor uses typical CISC architecture with **microprogrammed control**.

- The **instruction set** contained about **300 instructions** with **20** different **addressing modes**.
- The CPU in the VAX 8600 consisted of **two functional units** for concurrent execution of **integer and floating-point instructions.**
- The **unified cache** was used for holding both instructions and data.
- There were **16 GPRs** in the instruction unit.
- **Instruction pipelining** was built with six stages in the VAX 8600.

VAX 8600 Processor

- The **instruction unit prefetched and decoded instructions**, handled branching operations, and **supplied operands to the two functional units** in a pipelined fashion.
- A **translation lookaside buffer [TLB)** was used in the memory control unit **for fast generation of a physical address from a virtual address.**
- Both integer and floating-point units were pipelined.
- The **CPI of VAX 8600** instruction varied from **2 to 20** cycles. Because both multiply and divide instructions needs execution units for a large number of cycles.

The Motorola MC68040 microprocessor architecture



Captions:

IA = Instruction Address

DA = Data Address

EA = Effective Address

ATC = Addrress Translation Cache

MMU = Memory Management Unit

Motorola MC68040 microprocessor

Separate **instruction and data memory unit**, with a 4-Kbyte data cache, and a 4-Kbyte instruction cache, with separate ***memory management units*** **(MMUs)** supported by an ***address translation cache*** **(ATC),** equivalent to the TLB used in other systems.

- The processor implements 113 instructions using 16 general-purpose registers.

- **18-Addressing modes includes**:- register direct and indirect, indexing, memory indirect, program counter indirect, absolute, and immediate modes.

- The **instruction set** includes data movement, integer, BCD, and floating point arithmetic, logical, shifting, bit-field manipulation, cache maintenance, and multiprocessor communications, in addition to program and system control and memory management instructions

- The **integer unit** is organized in a **six-stage instruction pipeline**.

- The **floating-point unit** consists of **three pipeline stages** .

Motorola MC68040 microprocessor

Separate instruction and data buses are used to and from the instruction and data from memory units, respectively. **Dual MMUs allow interleaved fetch of instructions and data from the main memory.**

- Three simultaneous memory requests can he generated by the dual MMUs, including data operand read and write and instruction pipeline refill.

- **Snooping logic** is built into the memory units for monitoring bus events for cache invalidation.

- The complete memory management is provided with support for virtual demand paged operating system.

- Each of the two **ATCs has 64 entries** providing fast translation from virtual address to physical address.

Table 4.2 *Representative CISC Scalar Processors of year 1990*

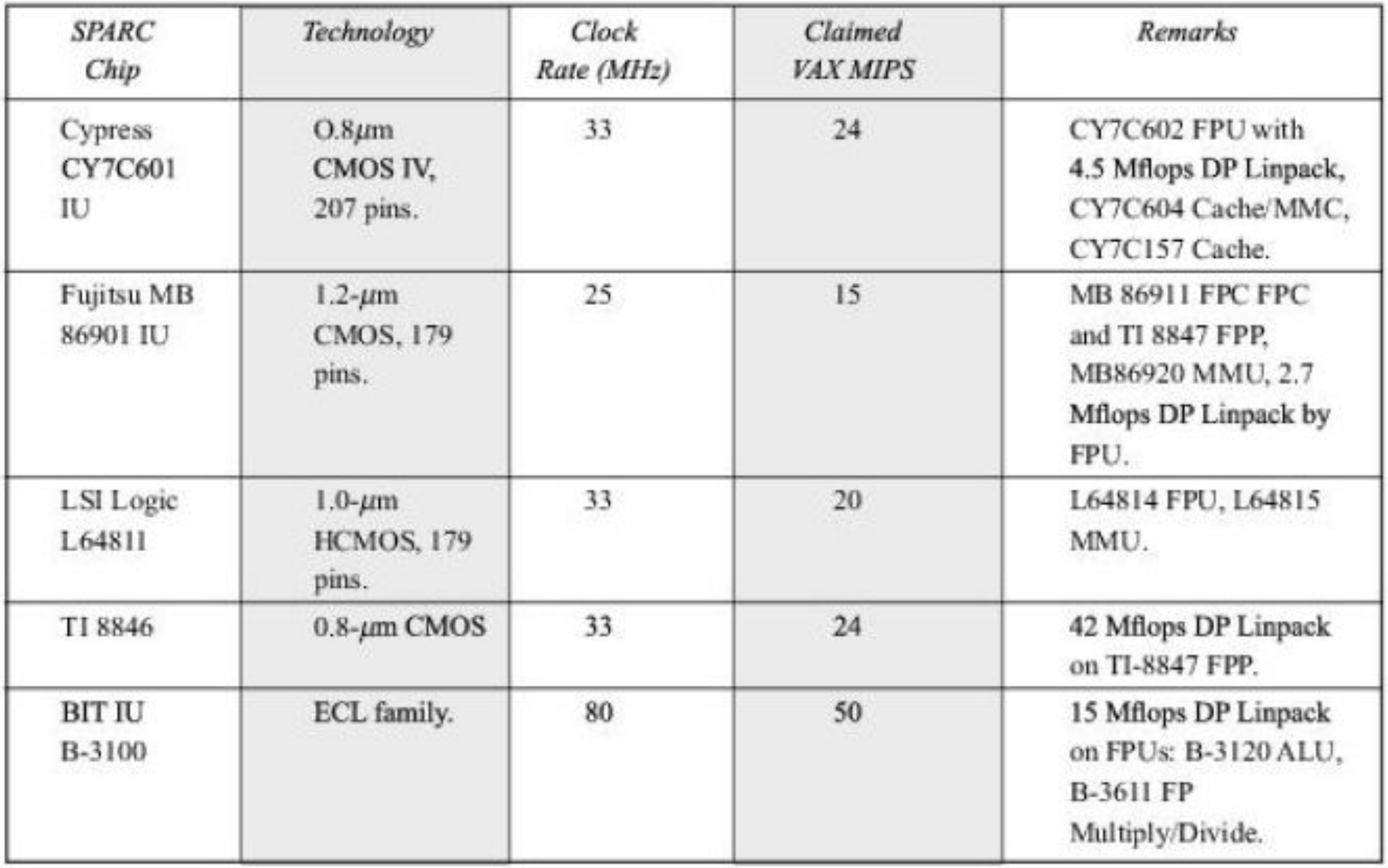
Feature	*Intel i486*	*Motorola MC68040*	*NS 32532*
Instruction-set size and word length	157 instructions, 32 bits.	113 instructions, 32 bits.	63 instructions, 32 bits.
Addressing modes	12	18	9
Integer unit and GPRs	32-bit ALU with 8 registers.	32-bit ALU with 16 registers.	32-bit ALU with 8 registers.
On-chip cache(s) and MMUs	8-KB unified cache for both code and data. with separate MMUs.	4-KB code cache 4-KB data cache	512-B code cache 1-KB data cache.
Floating-point unit, registers, and function units	On-chip with 8 FP registers adder, multiplier, shifter.	On-chip with 3 pipeline stages, 8 80-bit FP registers.	Off-chip FPU NS 32381, or WTL 3164.
Pipeline stages	5	6	4
Protection levels	4	2	2
Memory organization and TLB/ATC entries	Segmented paging with 4 KB/page and 32 entries in TLB.	Paging with 4 or 8 KB/page, 64 entries in each ATC.	Paging with 4 KB/page, 64 entries.
Technology, clock rate, packaging, and year introduced	CHMOS IV, 25 MHz, 33 MHz, 1.2M transistors, 168 pins, 1989.	0.8–μm HCMOS, 1.2 M transistors, 20 MHz, 40 MHz, 179 pins, 1990.	1.25–μm CMOS 370K transistors, 30 MHz, 175 pins, 1987.
Claimed performance	24 MIPS at 25 MHz,	20 MIPS at 25 MHz, 30 MIPS at 60 MHz.	15 MIPS at 30 MHz.

RISC Scalar Processors

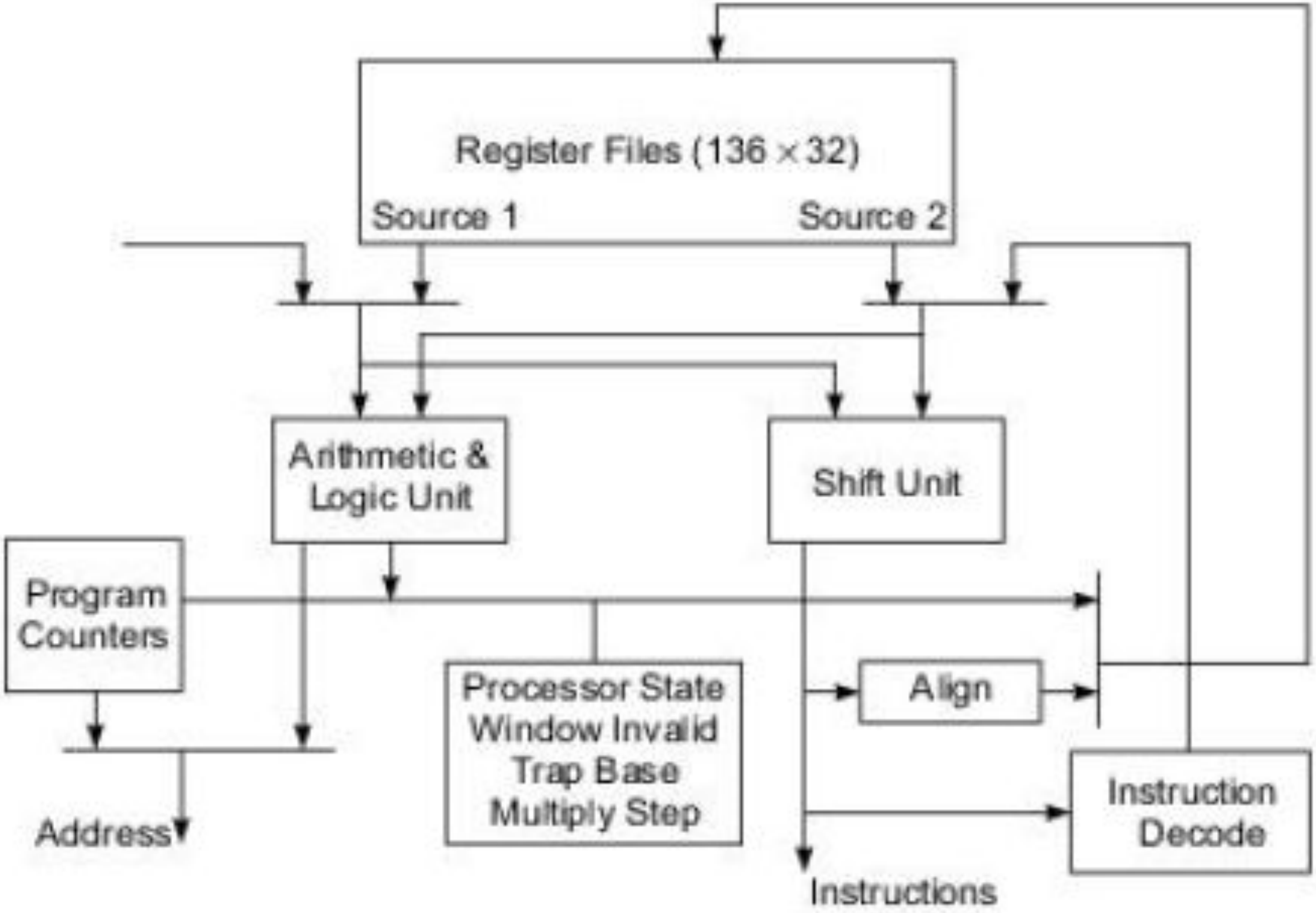
- **The Sun Microsystems SPARC architecture**
- **The lntel i860 processor architecture**

SPARC architecture

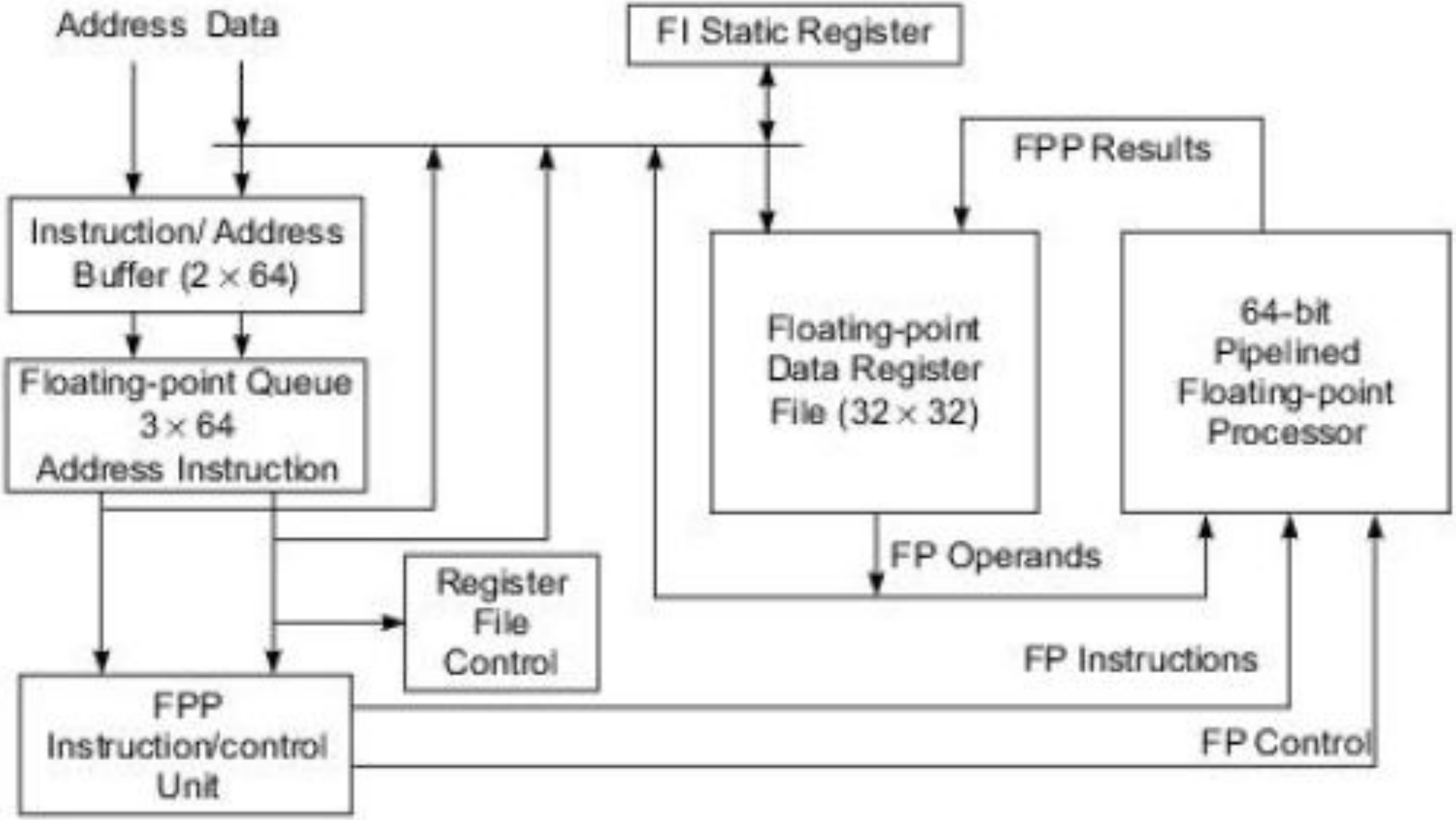
Table 4.4 *SPARC Implementations by Licensed Manufacturers (1990)*



SPARC Chip	*Technology*	*Clock Rate (MHz)*	*Claimed VAX MIPS*	*Remarks*
Cypress CY7C601 IU	O.8μm CMOS IV, 207 pins.	33	24	CY7C602 FPU with 4.5 Mflops DP Linpack, CY7C604 Cache/MMC, CY7C157 Cache.
Fujitsu MB 86901 IU	1.2-μm CMOS, 179 pins.	25	15	MB 86911 FPC FPC and TI 8847 FPP, MB86920 MMU, 2.7 Mflops DP Linpack by FPU.
LSI Logic L64811	1.0-μm HCMOS, 179 pins.	33	20	L64814 FPU, L64815 MMU.
TI 8846	0.8-μm CMOS	33	24	42 Mflops DP Linpack on TI-8847 FPP.
BIT IU B-3100	ECL family.	80	50	15 Mflops DP Linpack on FPUs: B-3120 ALU, B-3611 FP Multiply/Divide.



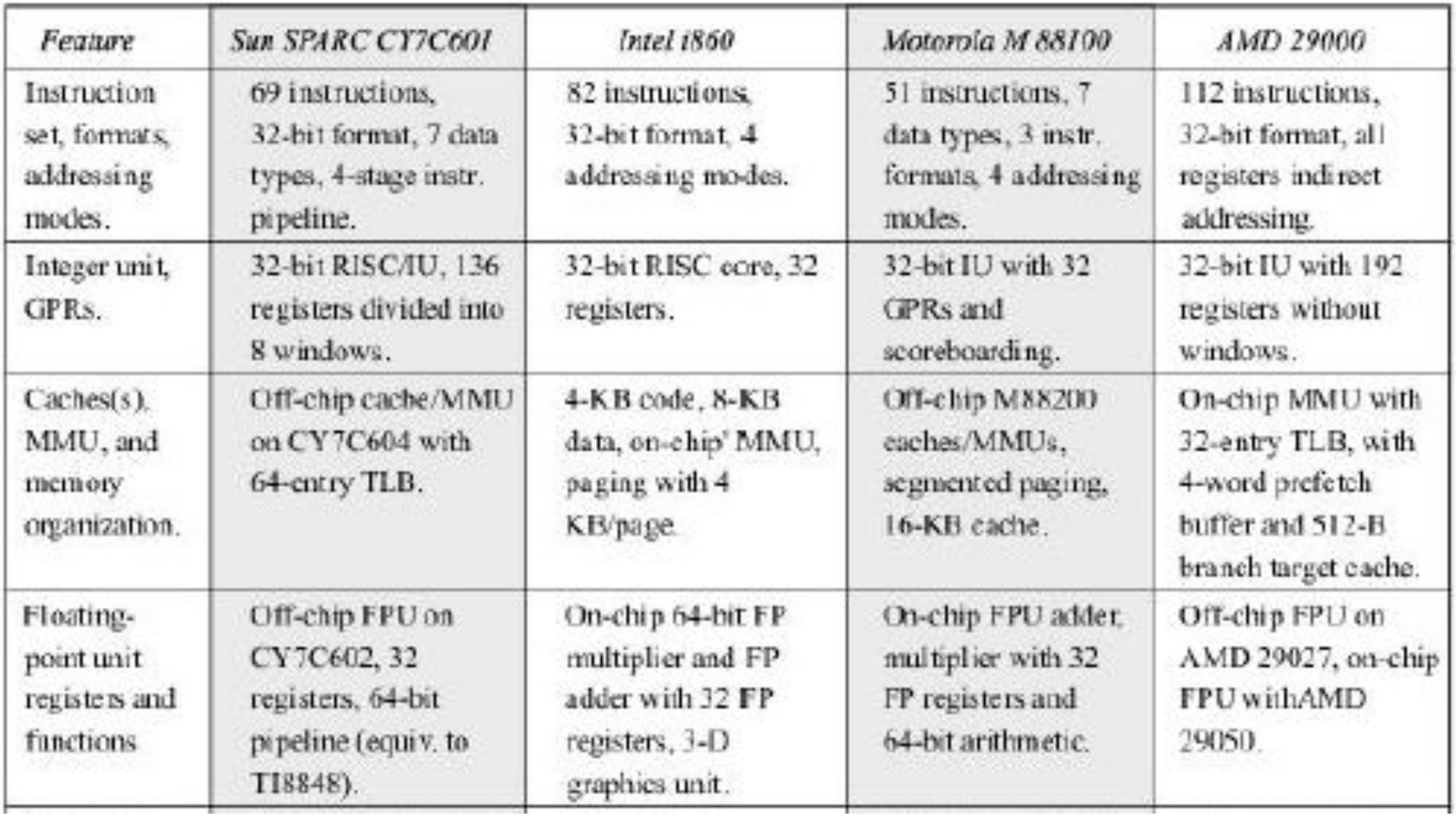
(a) The Cypress CY7C601 SPARC processor


Address
Data
FI Static Register
FPP Results
Instruction/ Address Buffer (2 × 64)
Floating-point Queue 3 × 64 Address Instruction
Floating-point Data Register File (32 × 32)
64-bit Pipelined Floating-point Processor
FP Operands
Register File Control
FP Instructions
FPP Instruction/control Unit
FP Control

(b) The Cypress CY7C602 floating-point unit

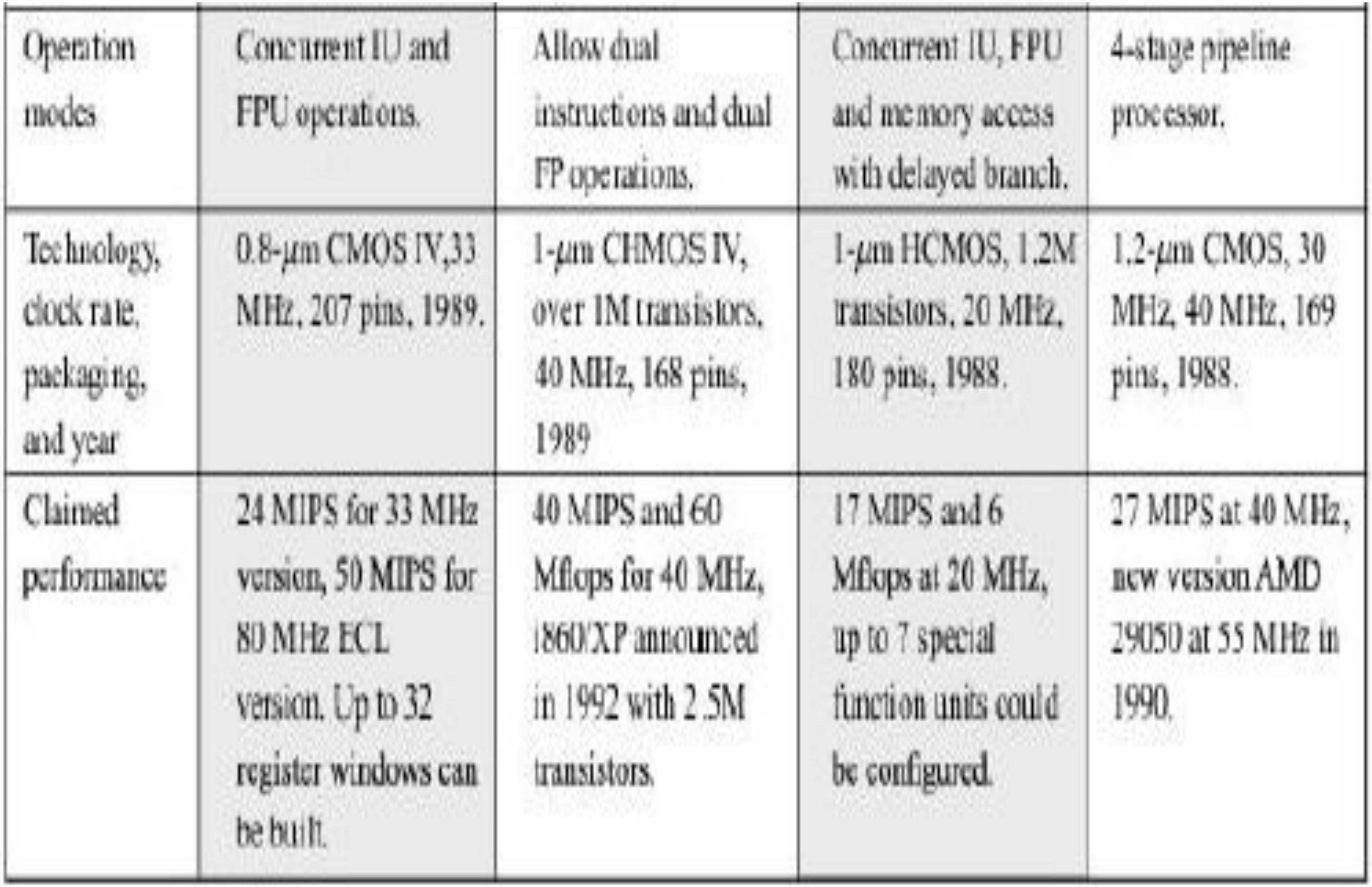
RISC Scalar Processors Cont…

Table 4.3 *Representative RISC Scalar Processors of year 199G*



Feature	*Sun SPARC CY7C601*	*Intel i860*	*Motorola M 88100*	*AMD 29000*
Instruction set, formats, addressing modes.	69 instructions, 32-bit format, 7 data types, 4-stage instr. pipeline.	82 instructions, 32-bit format, 4 addressing mo-des.	51 instructions, 7 data types, 3 instr. formats, 4 addressing modes.	112 instructions, 32-bit format, all registers indirect addressing.
Integer unit, GPRs.	32-bit RISC/IU, 136 registers divided into 8 windows.	32-bit RISC core, 32 registers.	32-bit IU with 32 GPRs and scoreboarding.	32-bit IU with 192 registers without windows.
Caches(s), MMU, and memory organization.	Off-chip cache/MMU on CY7C604 with 64-entry TLB.	4-KB code, 8-KB data, on-chip' MMU, paging with 4 KB/page.	Off-chip M88200 caches/MMUs, segmented paging, 16-KB cache.	On-chip MMU with 32-entry TLB, with 4-word prefetch buffer and 512-B branch target cache.
Floating-point unit registers and functions	Off-chip FPU on CY7C602, 32 registers, 64-bit pipeline (equiv. to TI8848).	On-chip 64-bit FP multiplier and FP adder with 32 FP registers, 3-D graphics unit.	On-chip FPU adder, multiplier with 32 FP registers and 64-bit arithmetic.	Off-chip FPU on AMD 29027, on-chip FPU withAMD 29050.

RISC Scalar Processors Cont…



Operation modes	Concurrent IU and FPU operations.	Allow dual instructions and dual FP operations.	Concurrent IU, FPU and memory access with delayed branch.	4-stage pipeline processor.
Technology, clock rate, packaging, and year	0.8-μm CMOS IV,33 MHz, 207 pins, 1989.	1-μm CHMOS IV, over 1M transistors, 40 MHz, 168 pins, 1989	1-μm HCMOS, 1.2M transistors, 20 MHz, 180 pins, 1988.	1.2-μm CMOS, 30 MHz, 40 MHz, 169 pins, 1988.
Claimed performance	24 MIPS for 33 MHz version, 50 MIPS for 80 MHz ECL version. Up to 32 register windows can be built.	40 MIPS and 60 Mflops for 40 MHz, i860/XP announced in 1992 with 2.5M transistors.	17 MIPS and 6 Mflops at 20 MHz, up to 7 special function units could be configured.	27 MIPS at 40 MHz, new version AMD 29050 at 55 MHz in 1990.

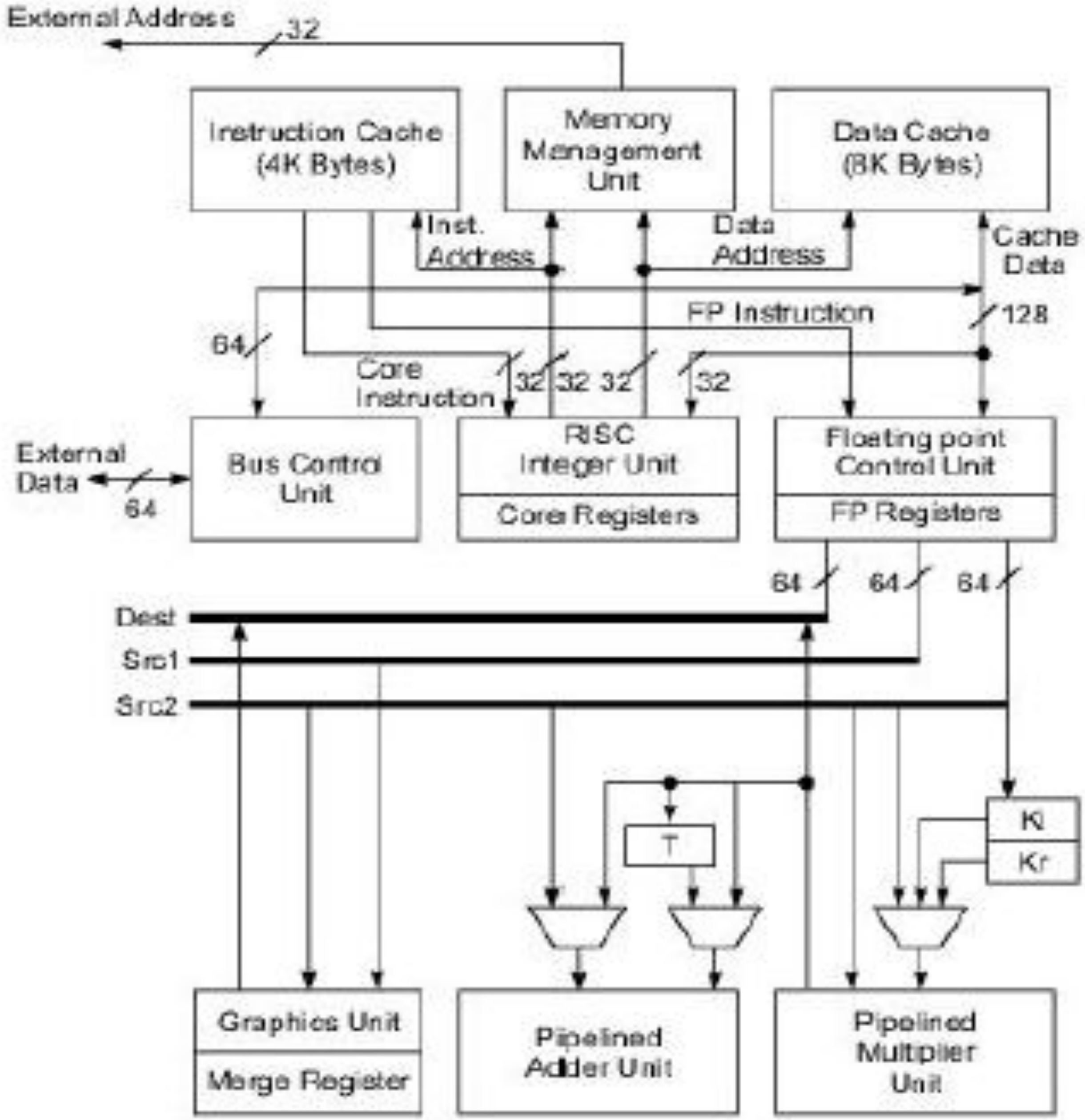


Fig. 4.9 Functional units and data paths of the Intel i860 RISC microprocessor (Courtesy of Intel Corporation, 1990)

Example 4.4 The Intel i860 processor architecture

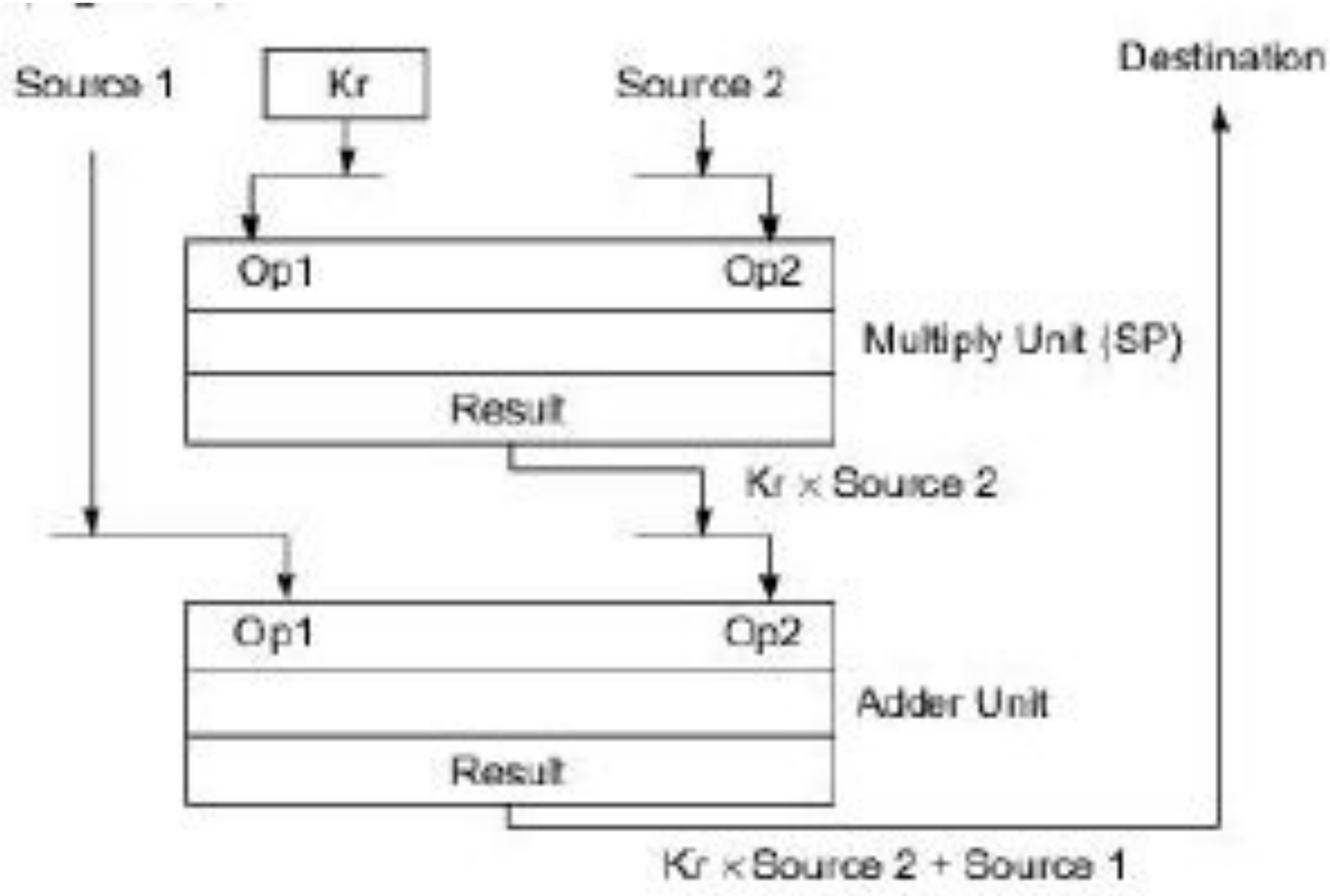


Fig. 4.10 Dual floating-point operations in the i860 processor

Superscalar And Vector Processors

- A CISC or a RISC scalar processor can be improved with a superscalar or vector architecture.
- Scalar processors are those executing one instruction per cycle.
- In a superscalar processor, multiple instructions are issued per cycle and multiple results are generated per cycle.

Superscalar And Vector Processors

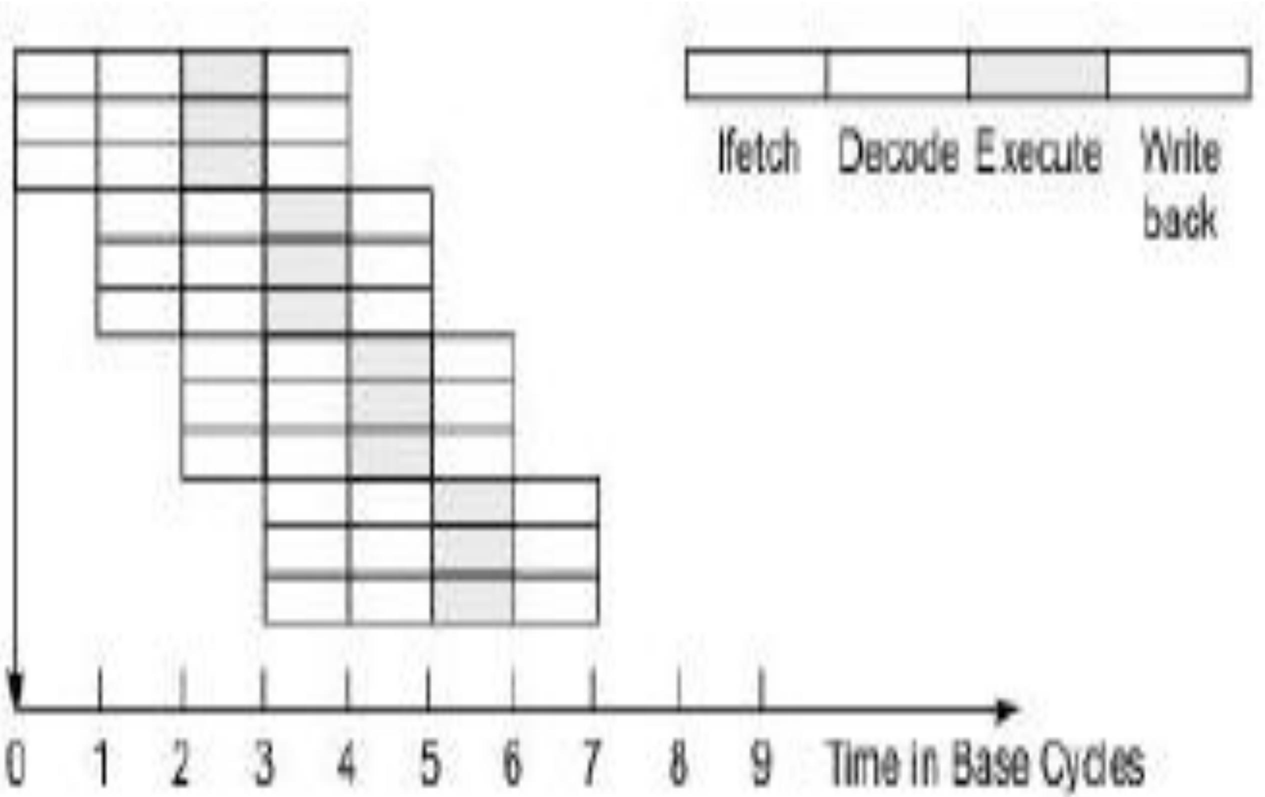


Fig. 4.11 A superscalar processor of degree m = 3

Table 4.5 *Representative Superscalar Processors (circa 1990)*

Feature	*Intel* *i960CA*	*IBM* *RS/6000*	*DEC Alpha* *21064*
Technology, clock rate, year	25 MHz 1986	1-μm CMOS technology, 30 MHz, 1990.	0.75-μm CMOS, 150 MHz, 431 pins, 1992.
Functional units and multiple instruction issues	Issue up to 3 instructions (register, memory, and control) per cycle, seven functional units available for concurrent use.	POWER architecture, issue 4 instructions (1 FXU, 1 FPU, and 2 ICU operations) per cycle.	Alpha architecture, issue 2 instructions per cycle, 64-bit IU and FPU, 128-bit data bus, and 34-bit address bus implemented in initial version.
Registers, caches, MMU, address space	1-KB I-cache, 1.5-KB RAM, 4-channel I/O with DMA, parallel decode, multiported registers.	32 32-bit GPRs, 8-KB I-cache, 64-KB D-cache with separate TLBs.	32 64-bit GPRs, 8-KB I-cache, 8-KB D-cache, 64-bit virtual space designed, 43-bit address space implemented in initial version.
Floating-point unit and functions	On-chip FPU, fast multimode interrupt, multitask control.	On-chip FPU 64-bit multiply, add, divide, subtract, IEEE 754 standard.	On-chip FPU, 32 64-bit FP registers, 10-stage pipeline, IEEE and VAX FP standards.
Claimed performance and remarks	30 VAX/MIPS peak at 25 MHz, real-time embedded system control, and multiprocessor applications.	34 MIPS and 11 Mflops at 25 MHz on POWER station 530.	300 MIPS peak and 150 Mflops peak at 150 MHz, multiprocessor and cache coherence support.

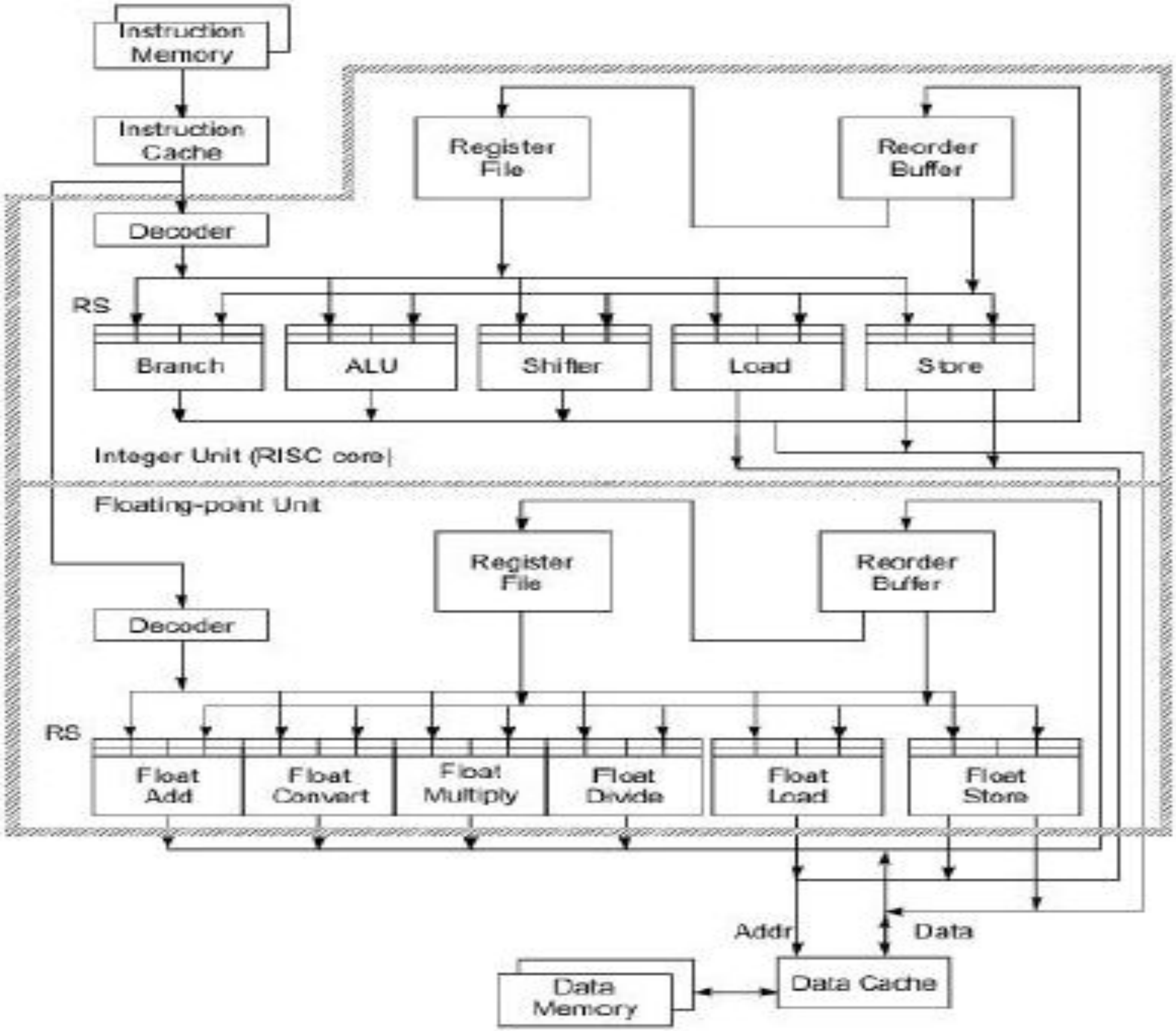


Fig. 4.12 A typical superscalar RISC processor architecture consisting of an integer unit and a floating-point unit (Courtesy of M. Johnson, 1991; reprinted with permission from Prentice-Hall, Inc.)

The IBM RS/6000 Architecture

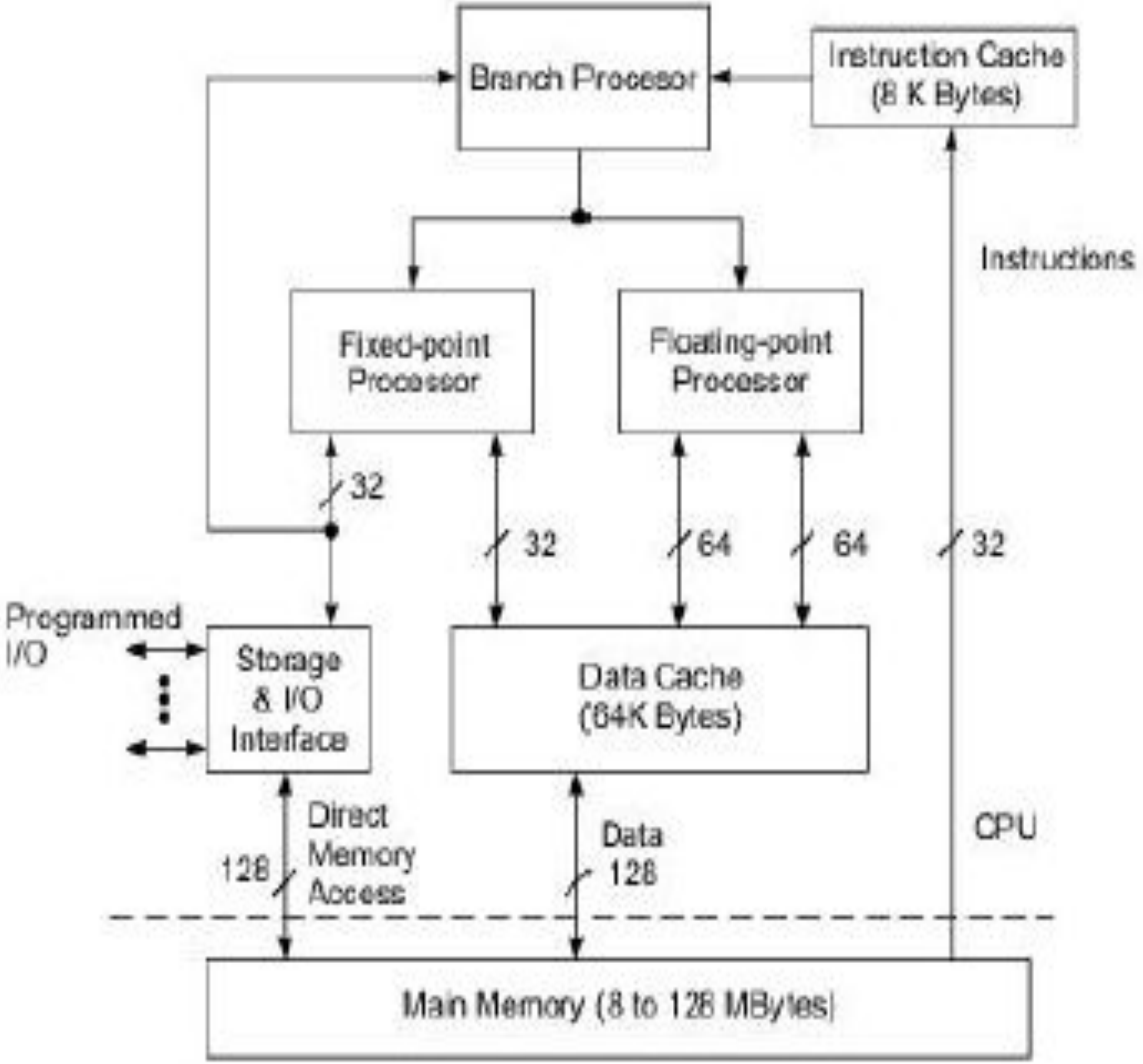
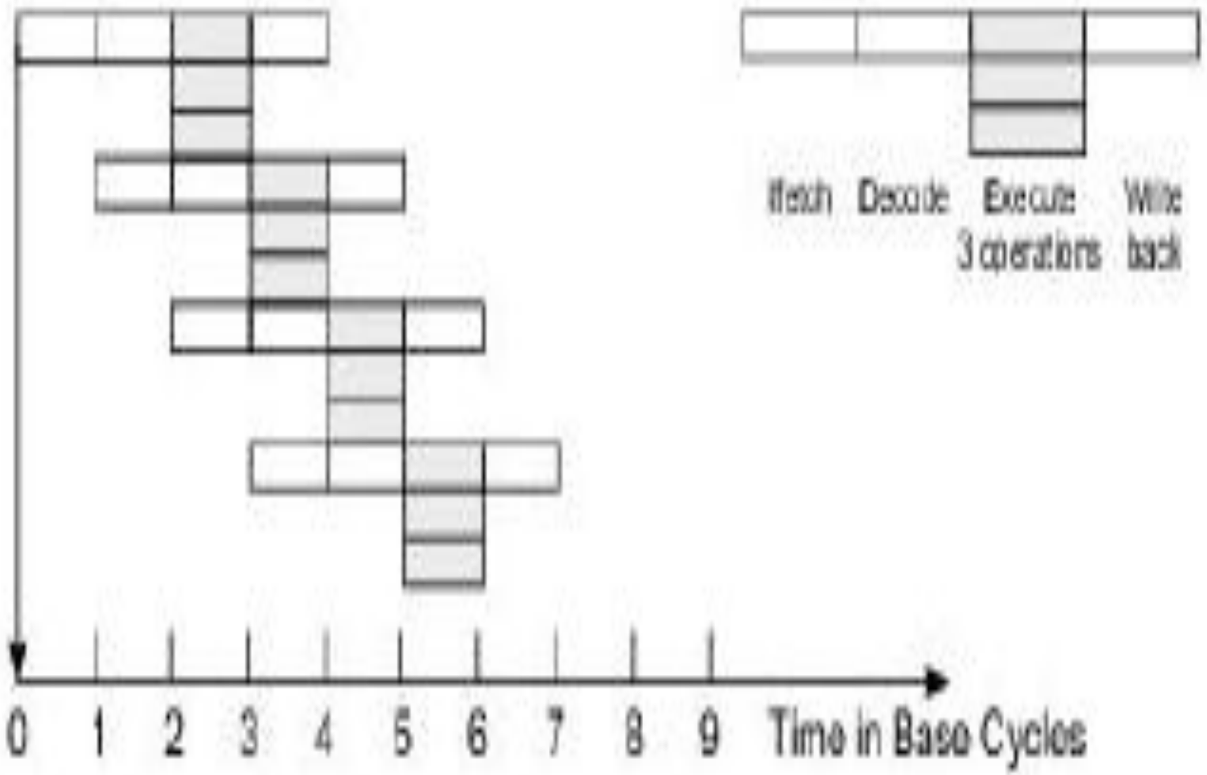


Fig. 4.13 The POWER architecture of the IBM RISC System/6000 superscalar processor (Courtesy of International Business Machines Corporation, 1990)

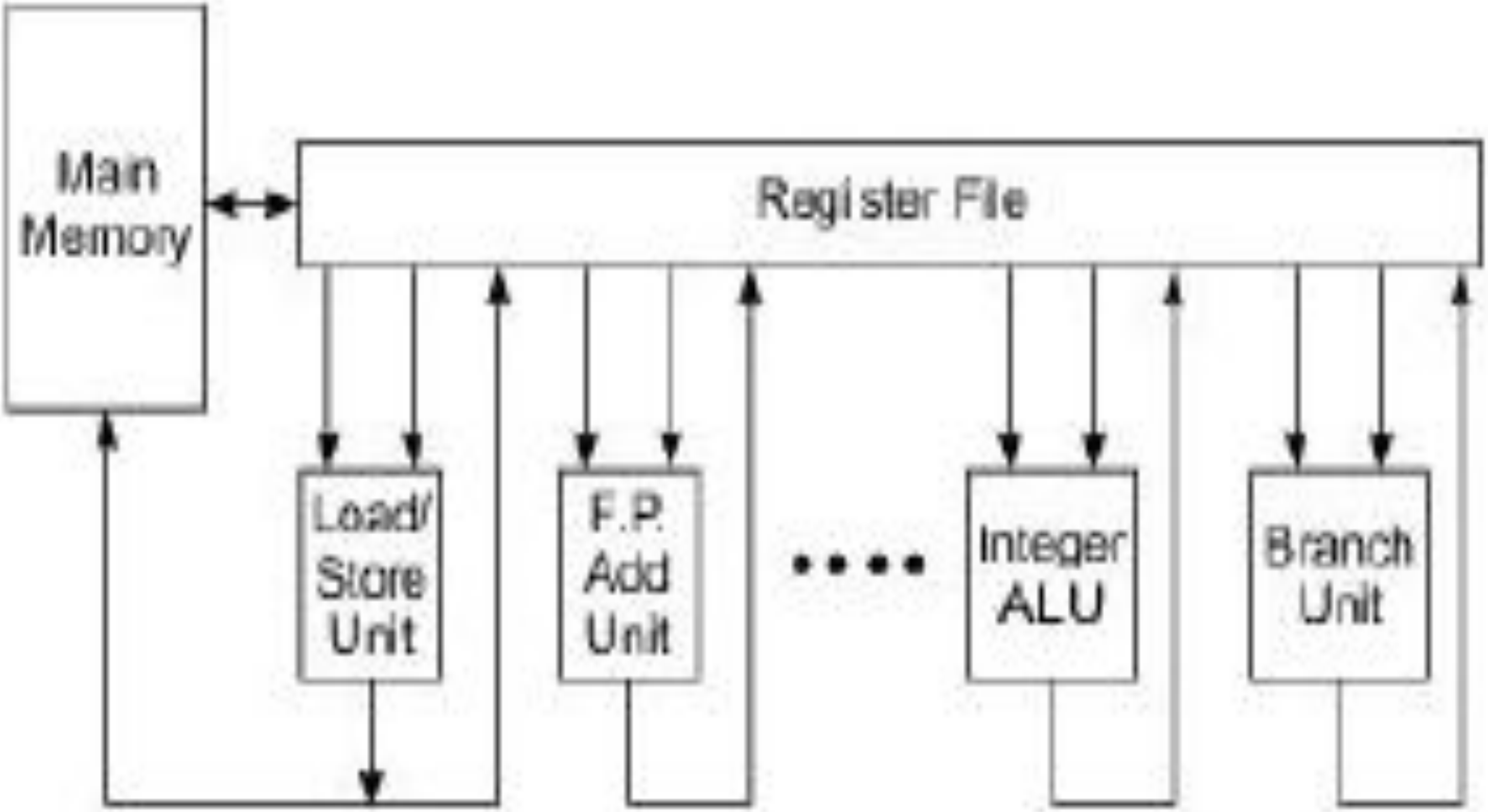
The VLIW Architecture

- A typical VLIW (very long instruction word) machine has instruction words hundreds of bits in length.



(b) VLIW execution with degree $m = 3$

Fig. 4.14 The architecture of a very long instruction word (VLIW) processor and its pipeline operations (Courtesy of Multiflow Computer, Inc., 1987)



Load/Store	FP Add	FP Multiply	Branch	• • •	Integer ALU

(a) A typical VLIW processor with degree $m = 3$

The VLIW Architecture

Pipelining in VLIW Processor:

- Each instruction specifies multiple operations.
- The effective CPI becomes 0.33 .
- VLIW machines behave much like superscalar machines with three differences:
- **First**, the decoding of VLIW instructions is easier than that of superscalar instructions.
- **Second**, the code density of the superscalar machine is better
- **Third**, a superscalar machine can be object-code-compatible with a large family of no n-parallel machines

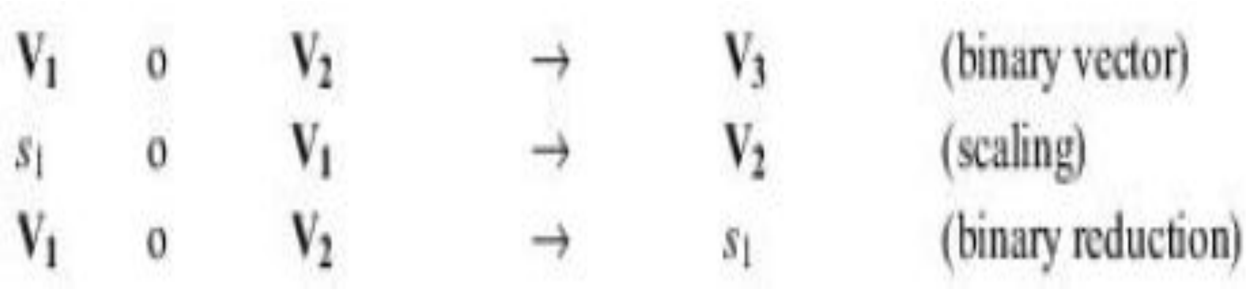
Vector and Symbolic Processors

- By definition, a vector processor is specially designed to perform vector computations.
- A vector instruction involves a large array of operands.
- A vector processor can assume either a register-in-register architecture or a memory-to-memory architecture.

Vector and Symbolic Processors

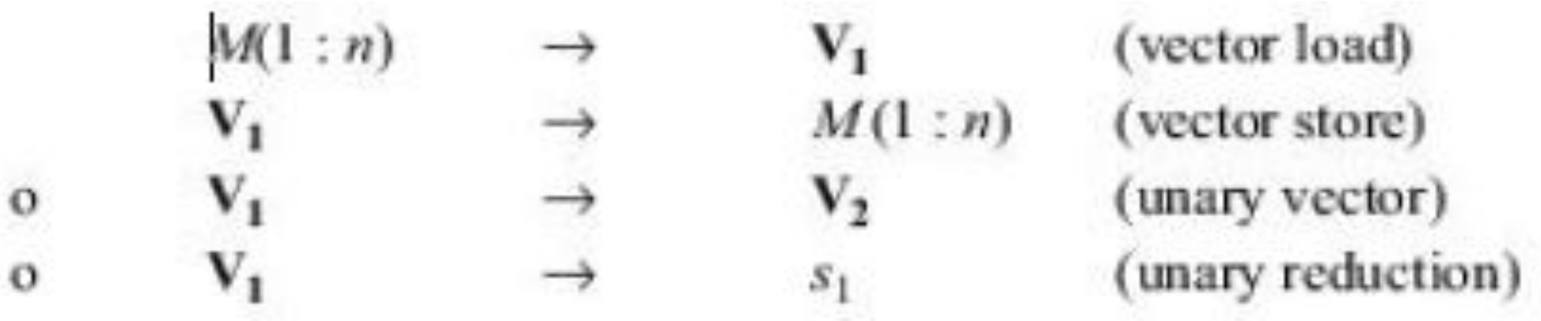
Vector Instructions

- Denote a vector register of length n as Vi, a scalar registers as si, and a memory array of length n as M(1:n).
- register-based vector operations
- vector operator is denoted by a small circle “o”.



V_1	o	V_2	→	V_3	(binary vector)
s_1	o	V_1	→	V_2	(scaling)
V_1	o	V_2	→	s_1	(binary reduction)

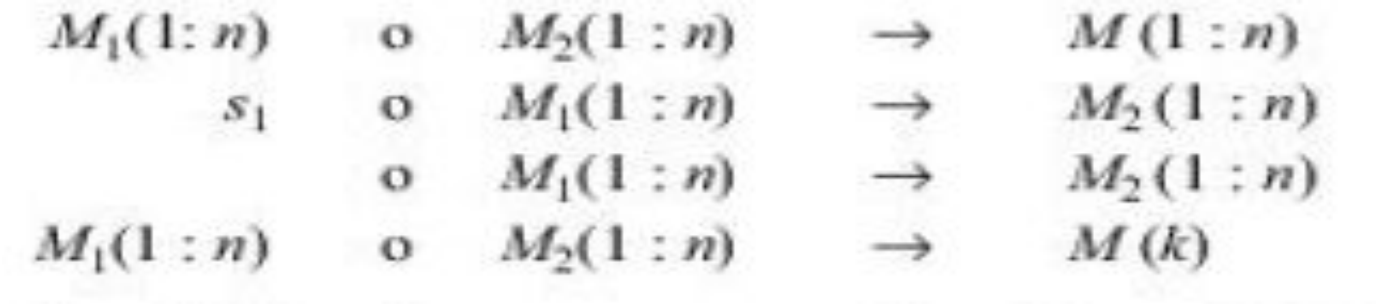
Vector and Symbolic Processors



	$M(1:n)$	$\rightarrow$	$\mathbf{V}_1$	(vector load)
	$\mathbf{V}_1$	$\rightarrow$	$M(1:n)$	(vector store)
o	$\mathbf{V}_1$	$\rightarrow$	$\mathbf{V}_2$	(unary vector)
o	$\mathbf{V}_1$	$\rightarrow$	s_1	(unary reduction)

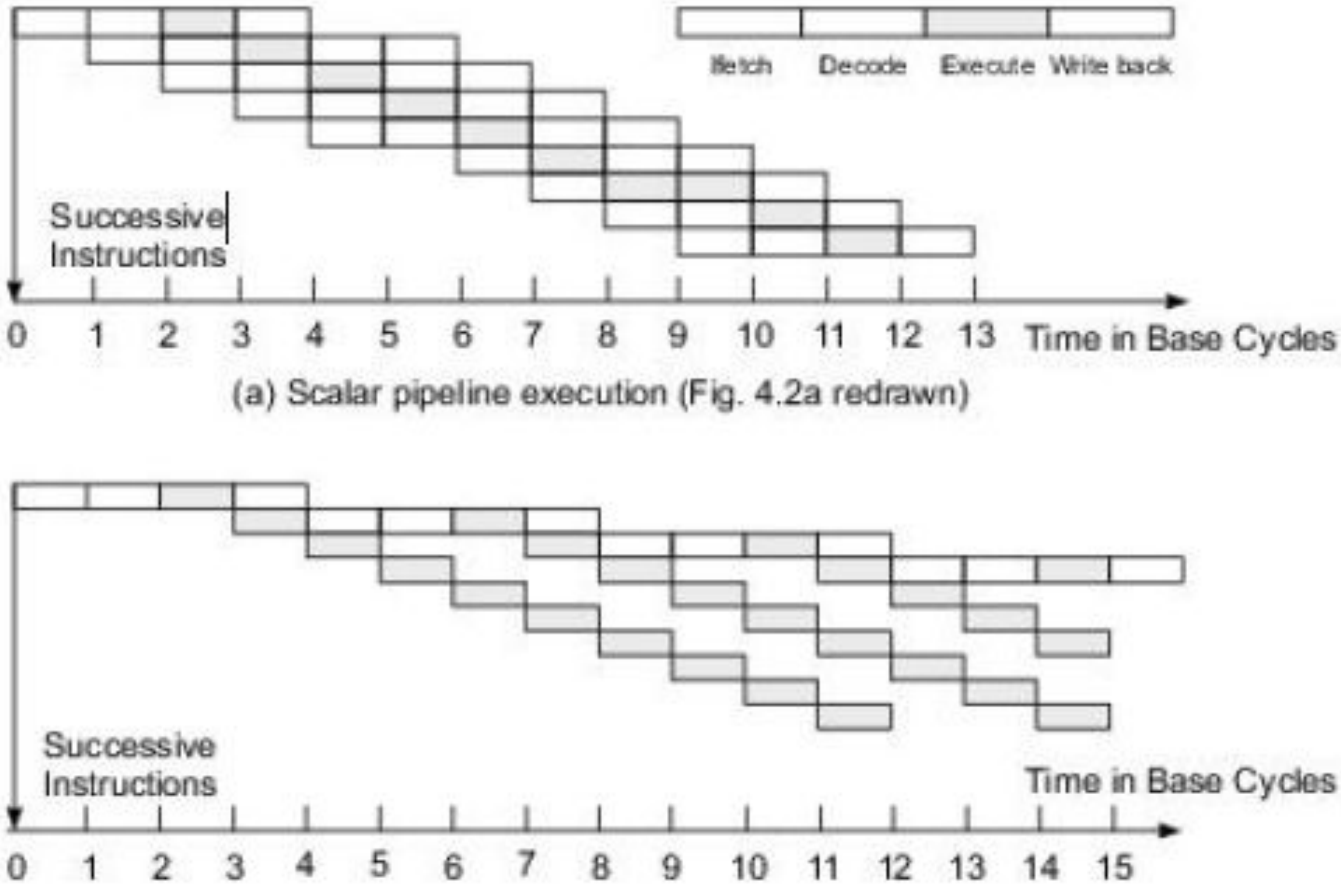
Vector and Symbolic Processors

- Memory-based vector operations are found in memory-to-memory vector :



$$\begin{array}{rcccl}
M_1(1:n) & \text{o} & M_2(1:n) & \rightarrow & M(1:n) \\
s_1 & \text{o} & M_1(1:n) & \rightarrow & M_2(1:n) \\
 & \text{o} & M_1(1:n) & \rightarrow & M_2(1:n) \\
M_1(1:n) & \text{o} & M_2(1:n) & \rightarrow & M(k)
\end{array}$$

- where M1(1:n) and M2(1:n) are two vectors of length n and M(k) denotes a scalar quantity stored in memory location k.



(a) Scalar pipeline execution (Fig. 4.2a redrawn)

(b) Vector pipeline execution

Fig. 4.15 Pipelined execution in a base scalar processor and in a vector processor, respectively (Courtesy of Jouppi and Wall; reprinted from Proc. ASPLOS, ACM Press, 1989)

Symbolic Processors

Symbolic Processor:

- Symbolic processing has been applied in many areas, including theorem proving, pattern recognition, expert systems, knowledge engineering, text retrieval cognitive science, and machine intelligence.

Symbolic Processors

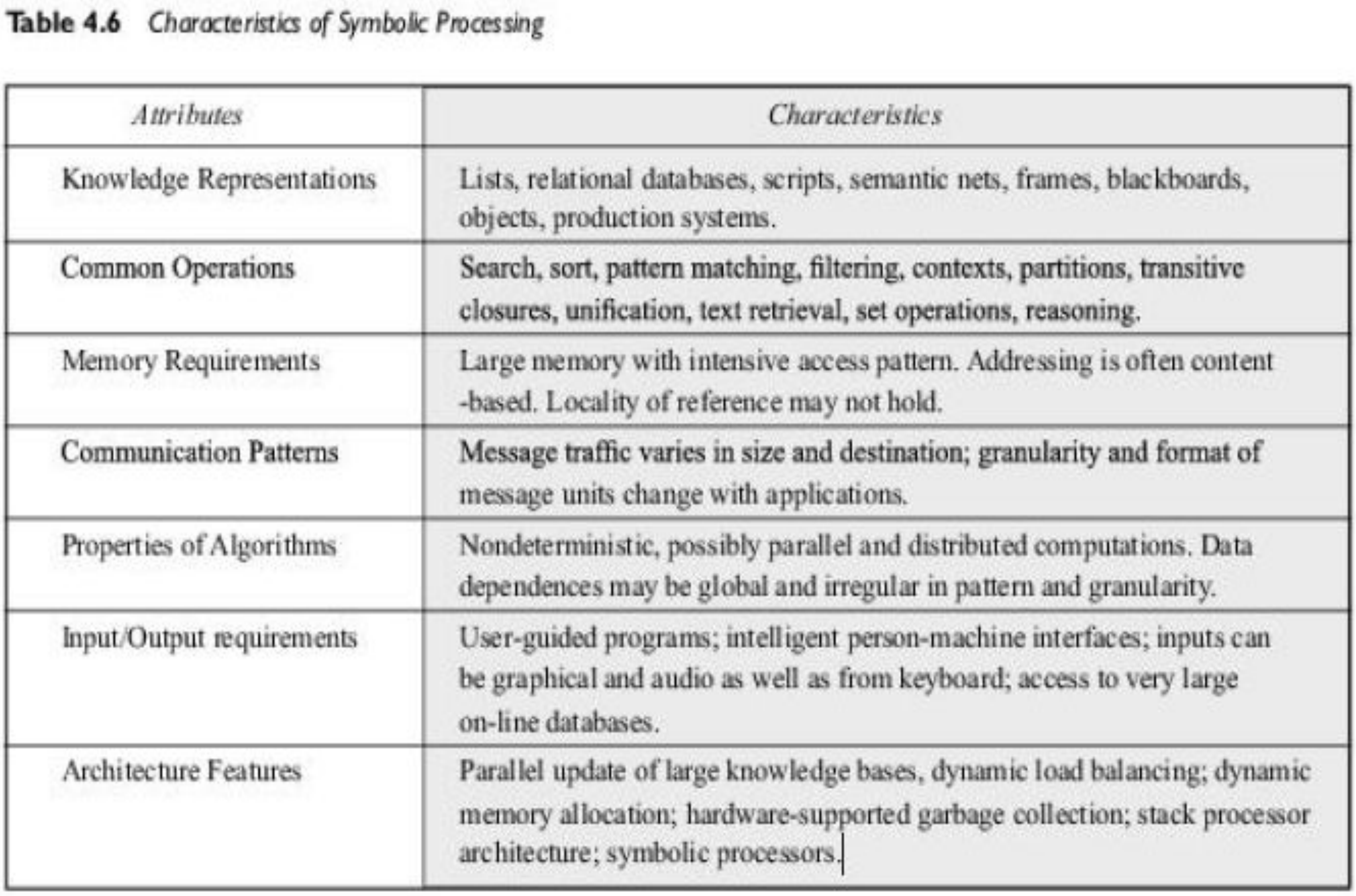


Table 4.6 *Characteristics of Symbolic Processing*

Attributes	*Characteristics*
Knowledge Representations	Lists, relational databases, scripts, semantic nets, frames, blackboards, objects, production systems.
Common Operations	Search, sort, pattern matching, filtering, contexts, partitions, transitive closures, unification, text retrieval, set operations, reasoning.
Memory Requirements	Large memory with intensive access pattern. Addressing is often content -based. Locality of reference may not hold.
Communication Patterns	Message traffic varies in size and destination; granularity and format of message units change with applications.
Properties of Algorithms	Nondeterministic, possibly parallel and distributed computations. Data dependences may be global and irregular in pattern and granularity.
Input/Output requirements	User-guided programs; intelligent person-machine interfaces; inputs can be graphical and audio as well as from keyboard; access to very large on-line databases.
Architecture Features	Parallel update of large knowledge bases, dynamic load balancing; dynamic memory allocation; hardware-supported garbage collection; stack processor architecture; symbolic processors.

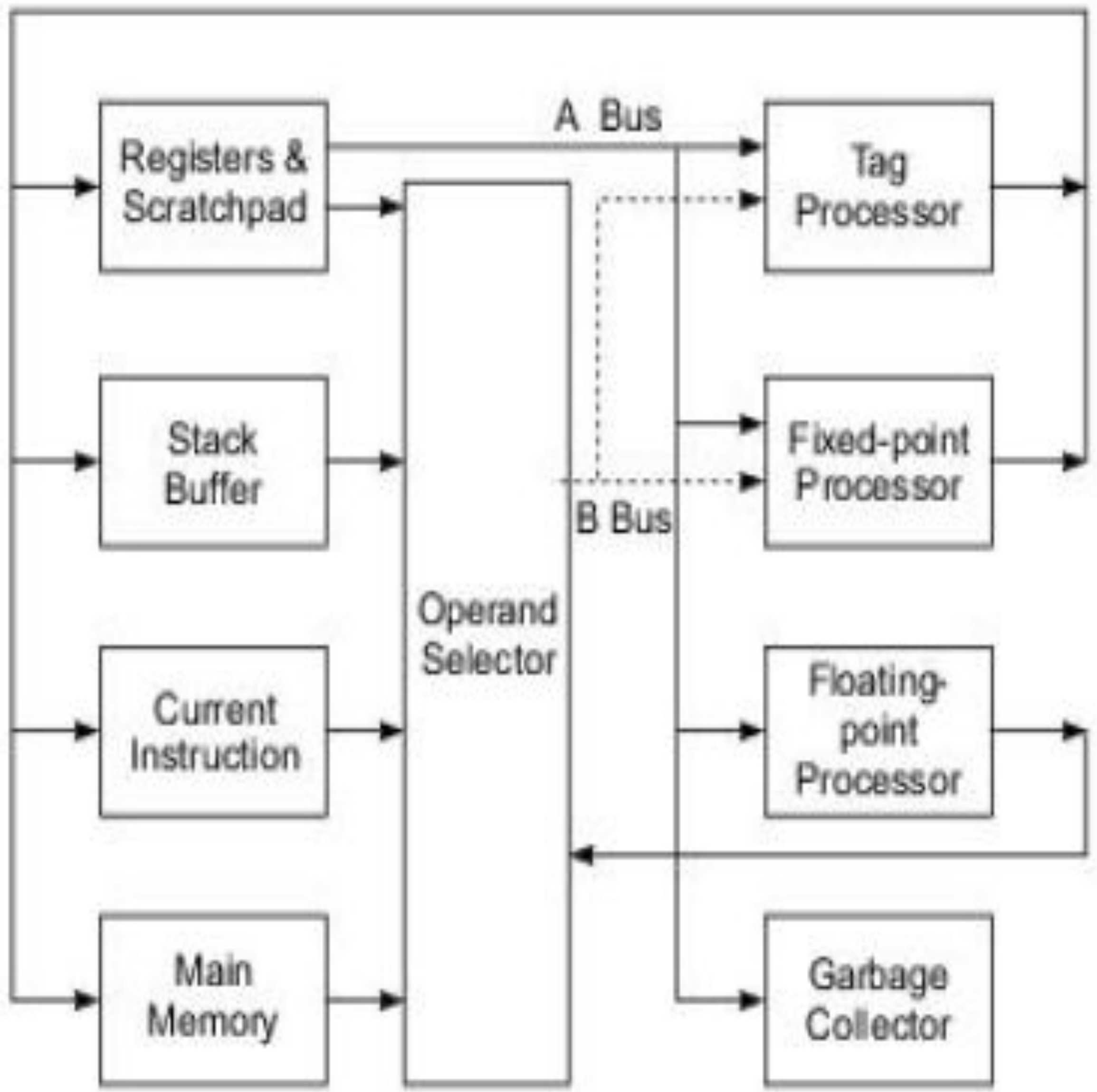


Fig. 4.16 The architecture of the Symbolics 3600 Lisp processor (Courtesy of Symbolics, Inc., 1985)

Memory Hierarchy Technology

- 1n a typical computer configuration, the cost of memory, disks, printers, and other peripherals often exceeds that of the processors.

Memory Hierarchy Technology

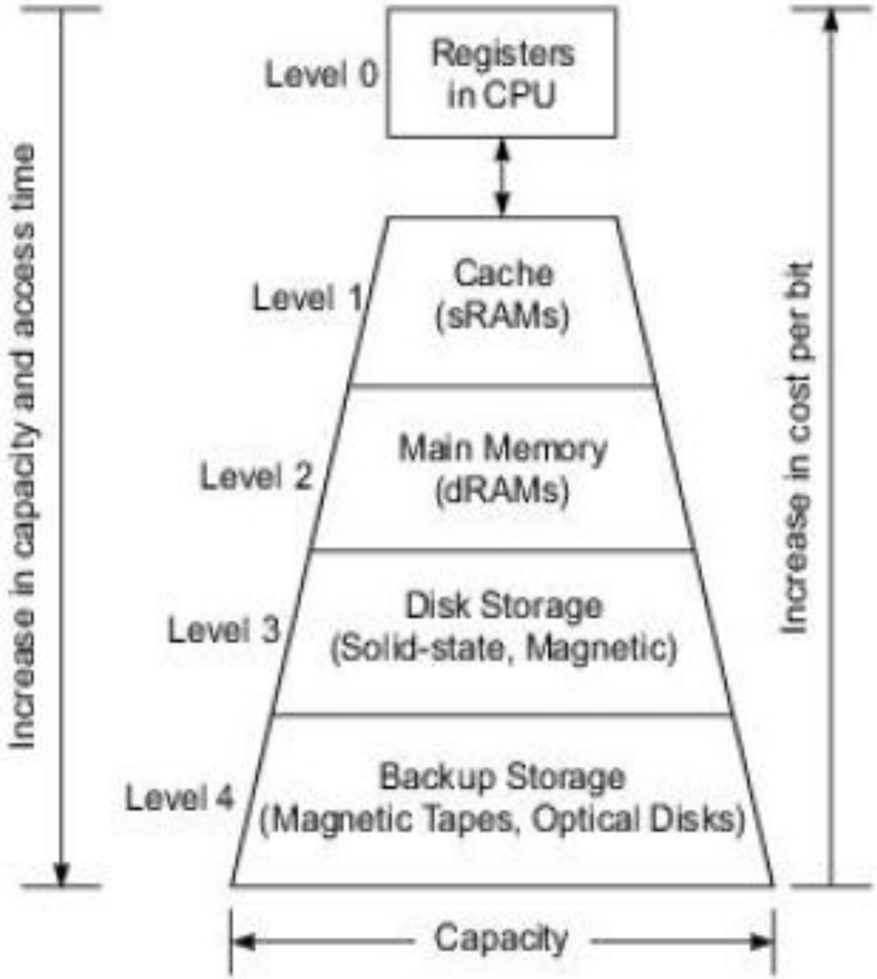

Level 0
Registers in CPU
Level 1
Cache (sRAMs)
Level 2
Main Memory (dRAMs)
Level 3
Disk Storage (Solid-state, Magnetic)
Level 4
Backup Storage (Magnetic Tapes, Optical Disks)
Increase in capacity and access time
Increase in cost per bit
Capacity

Fig. 4.17 A four-level memory hierarchy with increasing capacity and decreasing speed and cost from low to high levels

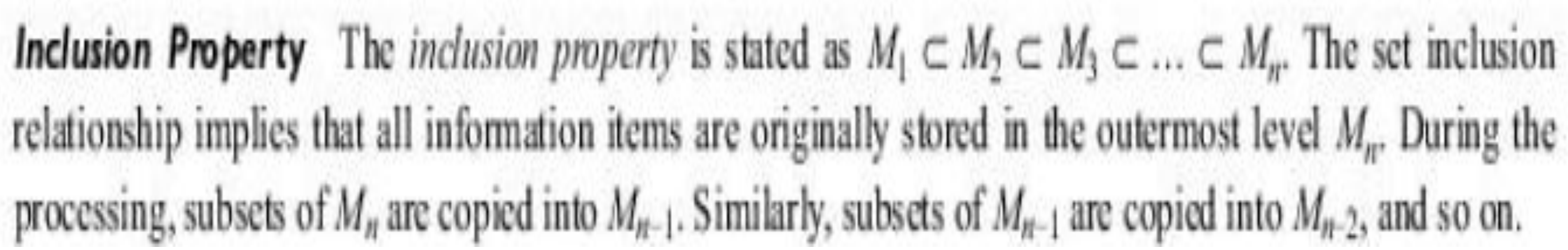
Memory Hierarchy Technology

- Access time (t_i)
- Memory size (s_i)
- Cost per byte(c_i)
- Transfer bandwidth (b_i)
- Unit of transfer (x_i)

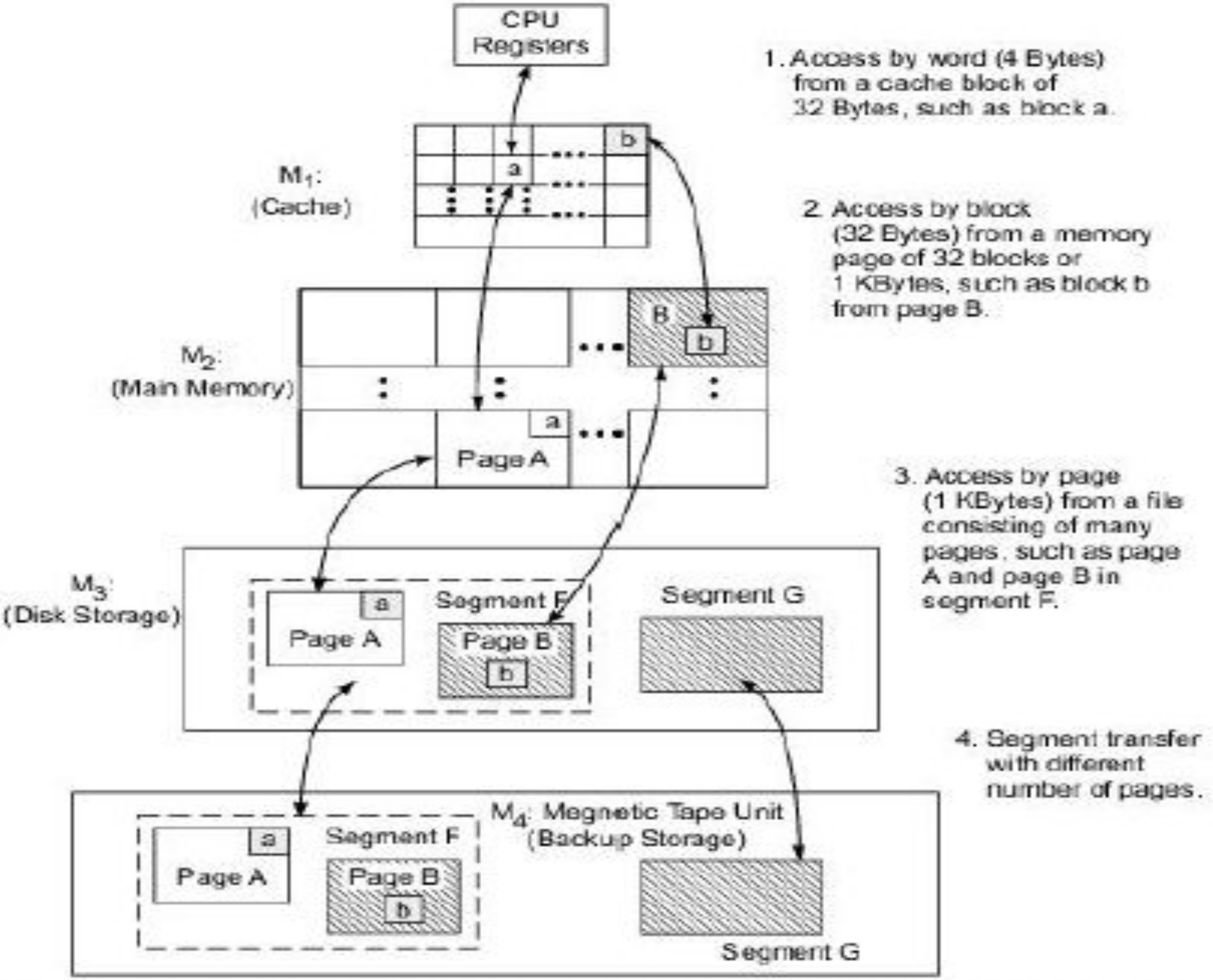
Inclusion, Coherence, And Locality

- Information stored in a memory hierarchy (M1,M2,….Mn) satisfies three important properties:
- inclusion
- coherence
- locality

Inclusion, Coherence, And Locality



Inclusion Property The *inclusion property* is stated as $M_1 \subset M_2 \subset M_3 \subset \ldots \subset M_n$. The set inclusion relationship implies that all information items are originally stored in the outermost level M_n. During the processing, subsets of M_n are copied into M_{n-1}. Similarly, subsets of M_{n-1} are copied into M_{n-2}, and so on.



1. Access by word (4 Bytes) from a cache block of 32 Bytes, such as block a.

2. Access by block (32 Bytes) from a memory page of 32 blocks or 1 KBytes, such as block b from page B.

3. Access by page (1 KBytes) from a file consisting of many pages, such as page A and page B in segment F.

4. Segment transfer with different number of pages.

Fig. 4.18 The inclusion property and data transfers between adjacent levels of a memory hierarchy

Inclusion, Coherence, And Locality

Locality of Reference:

- The memory hierarchy was developed based on a program behavior known as ***locality of references***.
- **Temporal Locality**:

Recently referenced items (instructions or data) are likely to be referenced again in the near future

Inclusion, Coherence, And Locality

2. Spatial Locality:

- This refers to the tendency for a process to access items whose addresses are near one another.
- For example, operations on tables or arrays involve accesses of a certain clustered area in the address space.

3. Sequential Locality:

- ln typical programs, the execution of instructions follows a sequential order (or the program order) unless branch instructions create out-of-order executions.

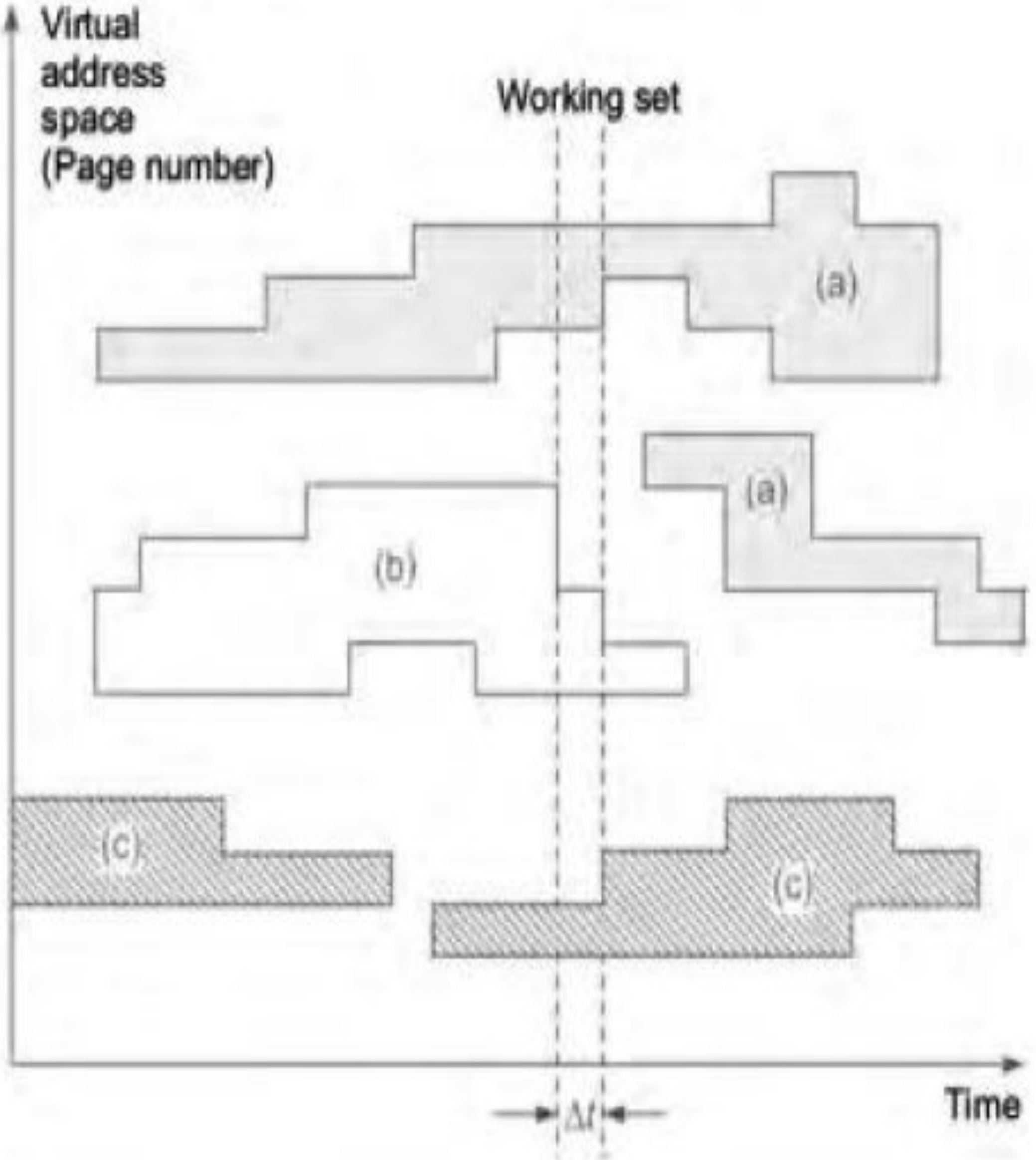


Fig. 4.19 Memory reference patterns in typical program trace experiments, where regions (a), (b), and (c) are generated with the execution of three software processes

Inclusion, Coherence, And Locality

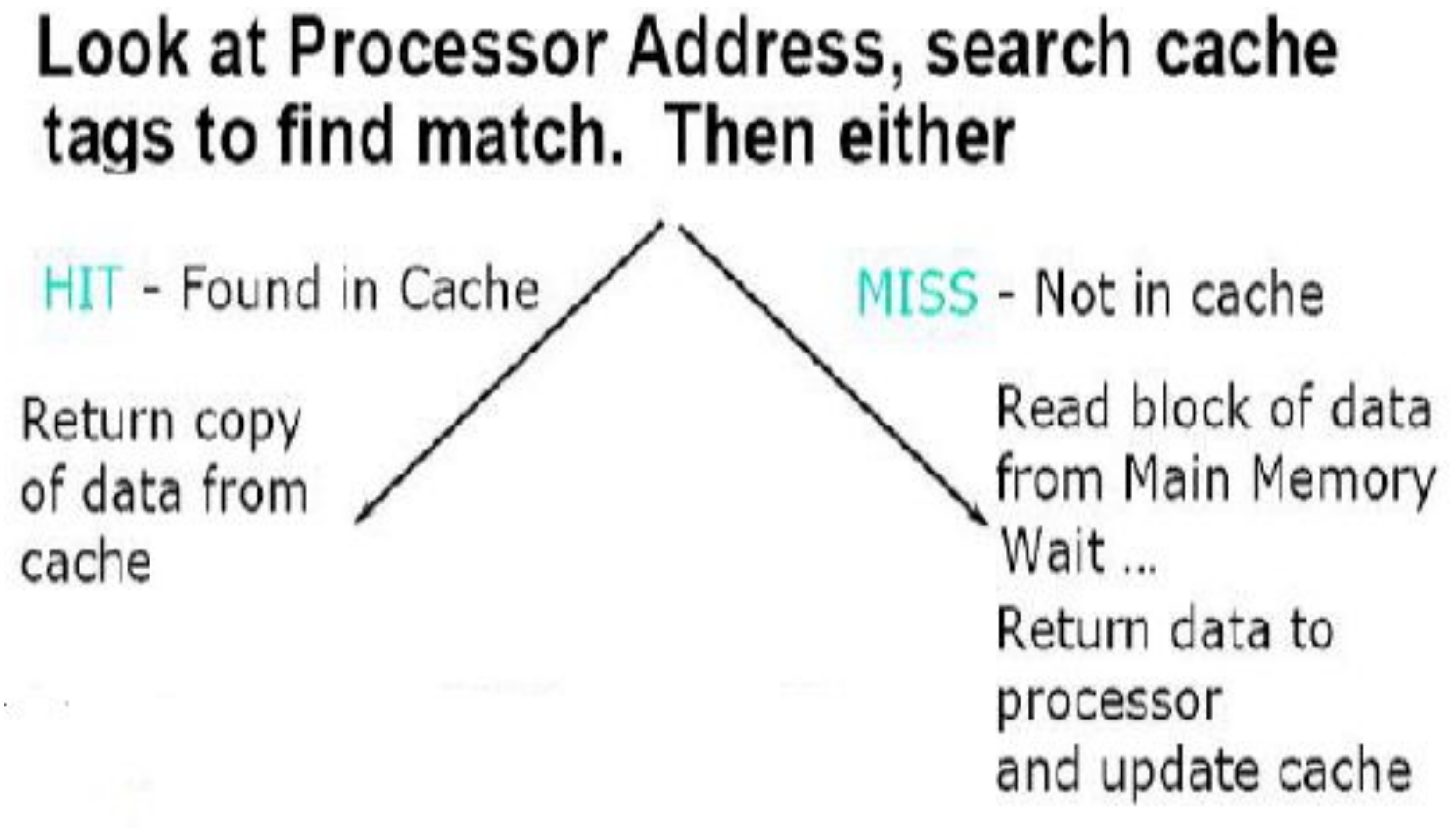
The Working Sets

- The subset of addresses (or pages) referenced within a given time window (t +Δt) is called the ***working set*** by Denning (1968).
- During the execution of a program, the working set changes slowly and maintains a certain degree of continuity

Coherence

- If a word is modified in the cache , all copies should be updated
- Write through
- Write back

Memory Capacity Planning



Look at Processor Address, search cache tags to find match. Then either

HIT - Found in Cache

Return copy of data from cache

MISS - Not in cache

Read block of data from Main Memory
Wait ...
Return data to processor
and update cache

Memory Capacity Planning

- effective access time T_{eff} to any level in the hierarchy.
- It depends on the hit ratios and access frequencies at successive levels.

Hit Ratios :

- When an information item is found in$M_i$we call it a hit, otherwise, a miss.
- M_i and M_{i-1}in a hierarchy, i= 1, 2,. . ., n.
- The hit ratio h_i; at M_i is the probability that an information item will be found in M_i .
- The miss ratio at $M_i$is defined as 1-h_i.

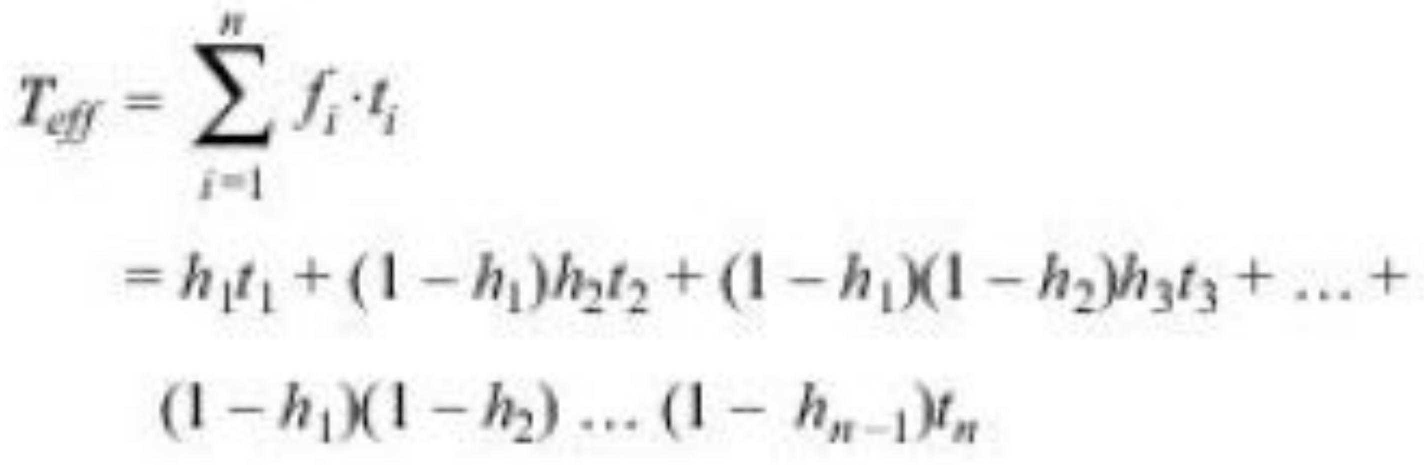
Memory Capacity Planning

Effective Access Time:

- block misses in the cache
- page faults in the main memory

Memory Capacity Planning

- Effective access time



$$T_{eff} = \sum_{i=1}^{n} f_i \cdot t_i$$

$$= h_1 t_1 + (1 - h_1) h_2 t_2 + (1 - h_1)(1 - h_2) h_3 t_3 + \ldots +$$

$$(1 - h_1)(1 - h_2) \ldots (1 - h_{n-1}) t_n$$

Memory Capacity Planning

Hierarchy Optimization

- The total cost of a memory hierarchy is estimated as follows:

Memory size (si)
Cost per byte(ci)

$$C_{total} = \sum_{I=1}^{n} C_i \cdot S_i$$

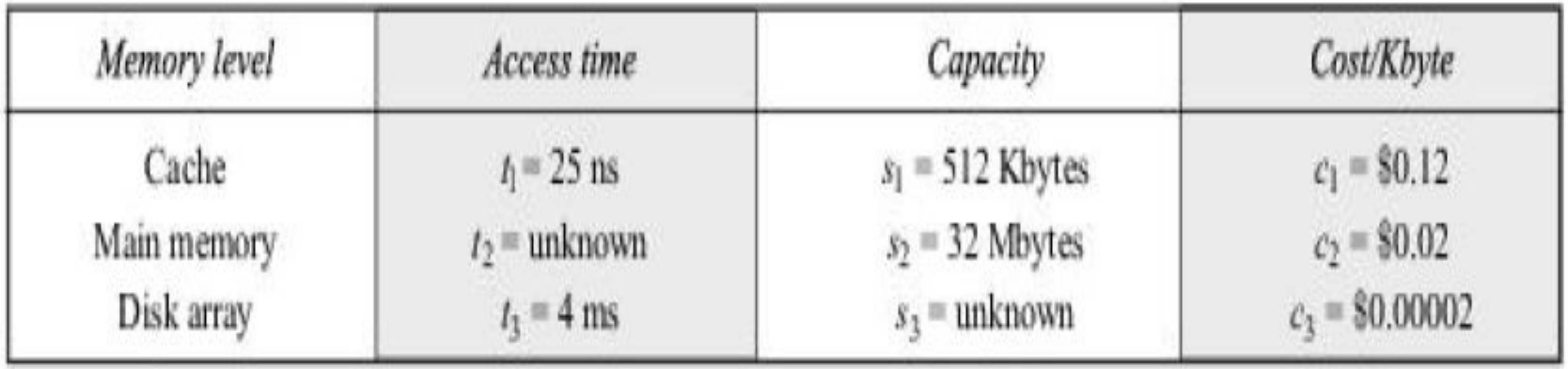
- This implies that the cost is distributed over n levels.
- Since $C_1 > C_2 > C_3 \ldots\ldots > C_{n,}$ we have to choose $S_1 < S_2 < S_3 < \ldots\ldots S_n$.

Memory Capacity Planning

- subject to the following constraints:
  - $s_i > 0, t_i > 0 \ for\ i = 1,2,3 \dots \dots, n$
    - $C_{total} = \sum_{I=1}^{n} C_i \cdot S_i < C_0$

Example 4.1 The Design Of A Memory Hierarchy

- Consider the design of a three-level memory hierarchy with the following specifications for



Memory level	*Access time*	*Capacity*	*Cost/Kbyte*
Cache	$t_1 = 25$ ns	$s_1 = 512$ Kbytes	$c_1 = \$0.12$
Main memory	$t_2 =$ unknown	$s_2 = 32$ Mbytes	$c_2 = \$0.02$
Disk array	$t_3 = 4$ ms	$s_3 =$ unknown	$c_3 = \$0.00002$

The design goal is to achieve an effective memory-access time t = 850 ns with a cache hit ratio h1 = 0.93 and a hit ratio h2 = 0.99 in main memory.

Example 4.1 The Design Of A Memory Hierarchy

- Also, the total cost of the memory hierarchy is upper-bounded by $1,500.
- The memory hierarchy cost is calculated as:

$$C= c_1 s_1 + c_2 s_2 + c_3 s_3 \le 1500$$

0.12*512+0.02*32*10^3+ s_3 *0.00002 $\le$ 1500

61.44+640+ s_3 *0.00002 $\le$ 1500

$s_3 = 39.8$

- The effective memory access
- time is calculated as :

Teff=h1t1+(1-h1)h2t2+(1-h1)(1-h2)h3t3

=0.93*25*10^-9

+0.07*0.99*t2

+0.07*0.01*1*4*10^-3

Teff =850*10^-9