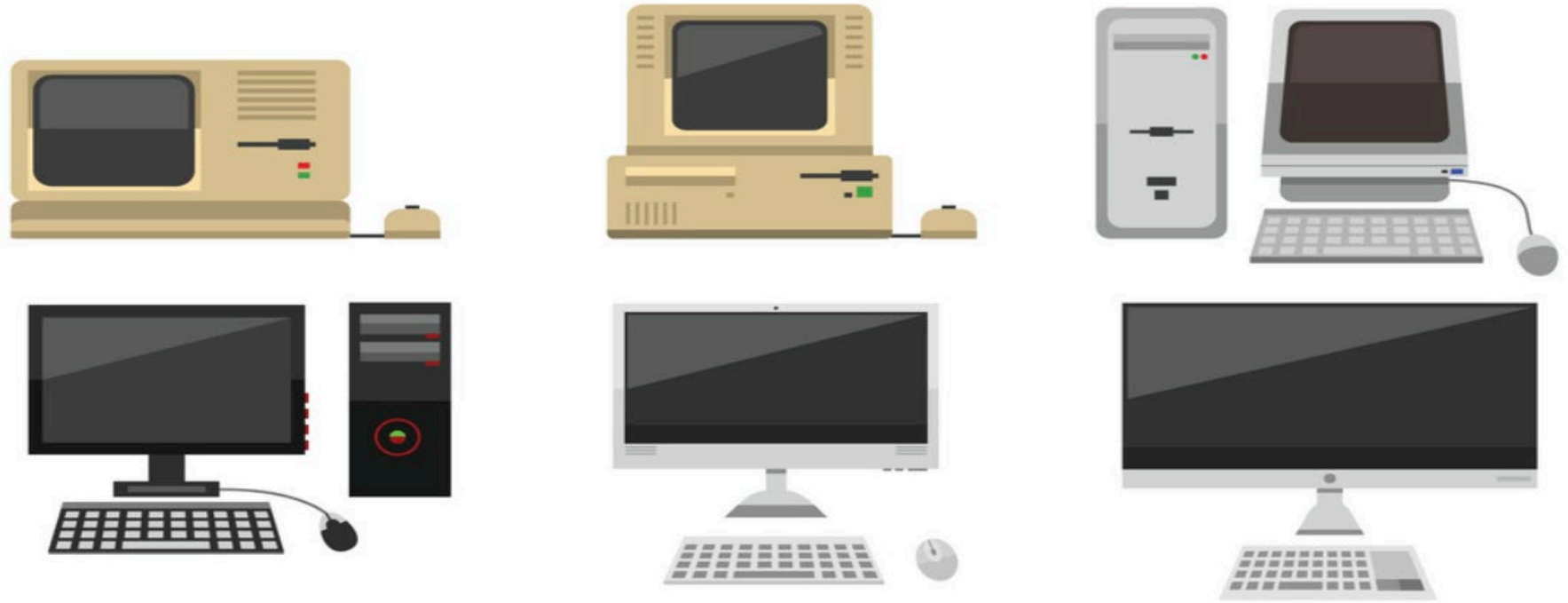
Chapter 2

Computer Evolution and Performance



Yousef Awwad Daraghmi

These slides are based on the book:
Computer Architecture and Organization Designing for Performance, William Stallings, 8th edition

The slides may include materials (questions, examples, case studies) from other resources

Answer to questions

- Memory size vs CPU speed

ENIAC - background

- **E**lectronic **N**umerical **I**ntegrator And **C**omputer
- Eckert and Mauchly
- University of Pennsylvania
- Trajectory tables for weapons
- Started 1943
- Finished 1946
  - Too late for war effort
- Used until 1955

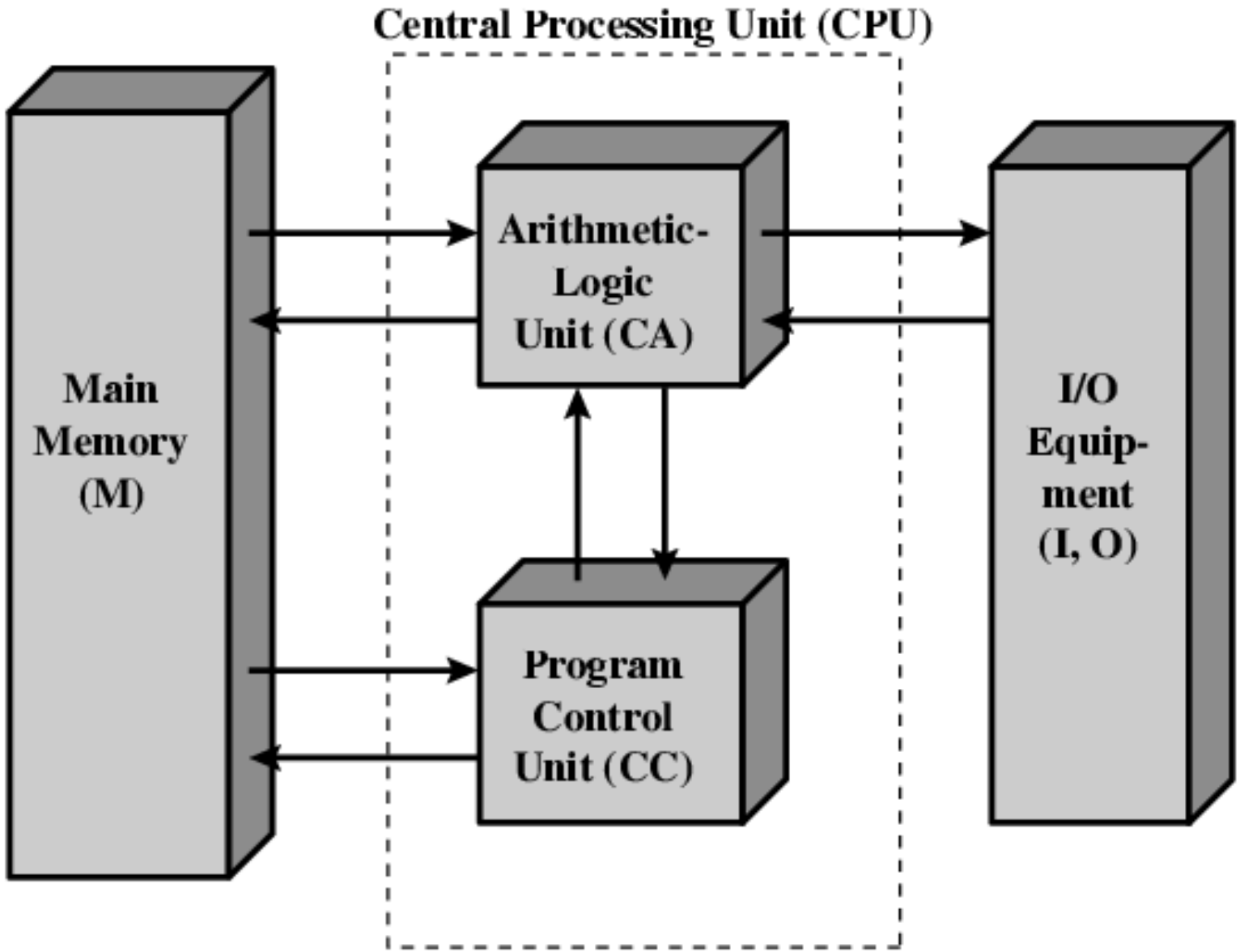
ENIAC - details

- Decimal (not binary)
- 20 accumulators of 10 digits
- Programmed manually by switches
- 18,000 vacuum tubes
- 30 tons
- 15,000 square feet
- 140 kW power consumption
- 5,000 additions per second

von Neumann/Turing

- Stored Program concept
- Main memory storing programs and data
- ALU operating on binary data
- Control unit interpreting instructions from memory and executing
- Input and output equipment operated by control unit
- Princeton Institute for Advanced Studies
  - —IAS
- Completed 1952

Structure of von Neumann machine

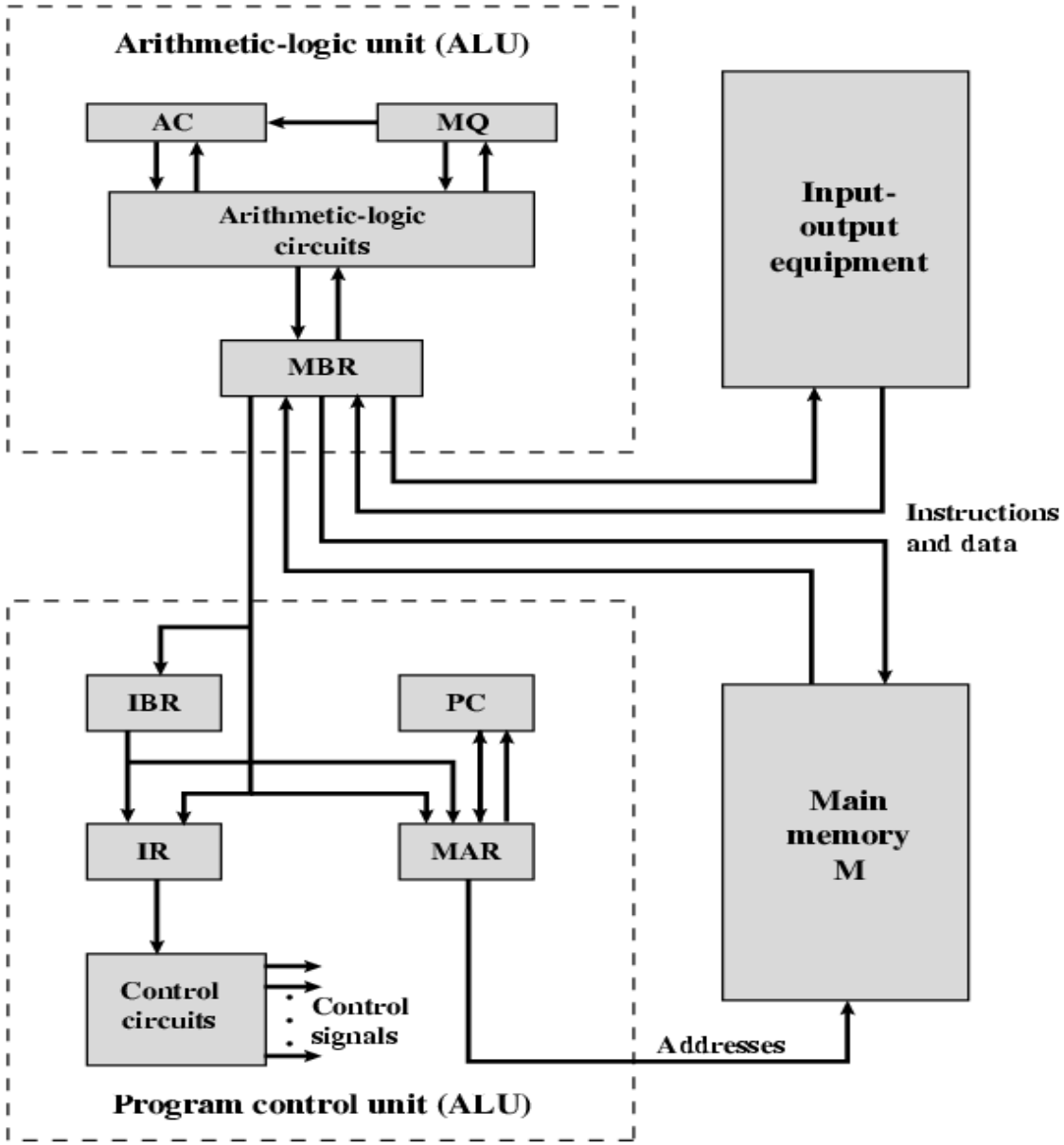


IAS - details

- First Electronic computer
- The IAS machine was a binary computer with a 40 bit word, storing two 20 bit instructions in each word. The memory was 1024 words. Negative numbers were represented in "two's complement" format.
- Set of registers (**storage in CPU**)
  - **Memory Buffer Register (MBR):** Contains a word to be stored in memory or sent to the I/O unit, or is used to receive a word from memory or from the I/O unit.
  - **Memory Address Register (MAR):** Specifies the address in memory of the word to be written from or read into the MBR.
  - **Instruction Register (IR):** Contains the 8-bit **opcode** instruction being executed.
  - **Instruction Buffer Register (IBR):** Employed to hold temporarily the right hand instruction from a word in memory.
  - **Program Counter (PC):** Contains **the address of the next instruction-pair** to be fetched from memory.
  - **Accumulator (AC) and Multiplier Quotient (MQ):** Employed to hold temporarily operands and results of ALU operations. For example, the result of multiplying two 40-bit numbers is an 80-bit number; the most significant 40 bits are stored in the AC and the least significant in the MQ.

IAS - details

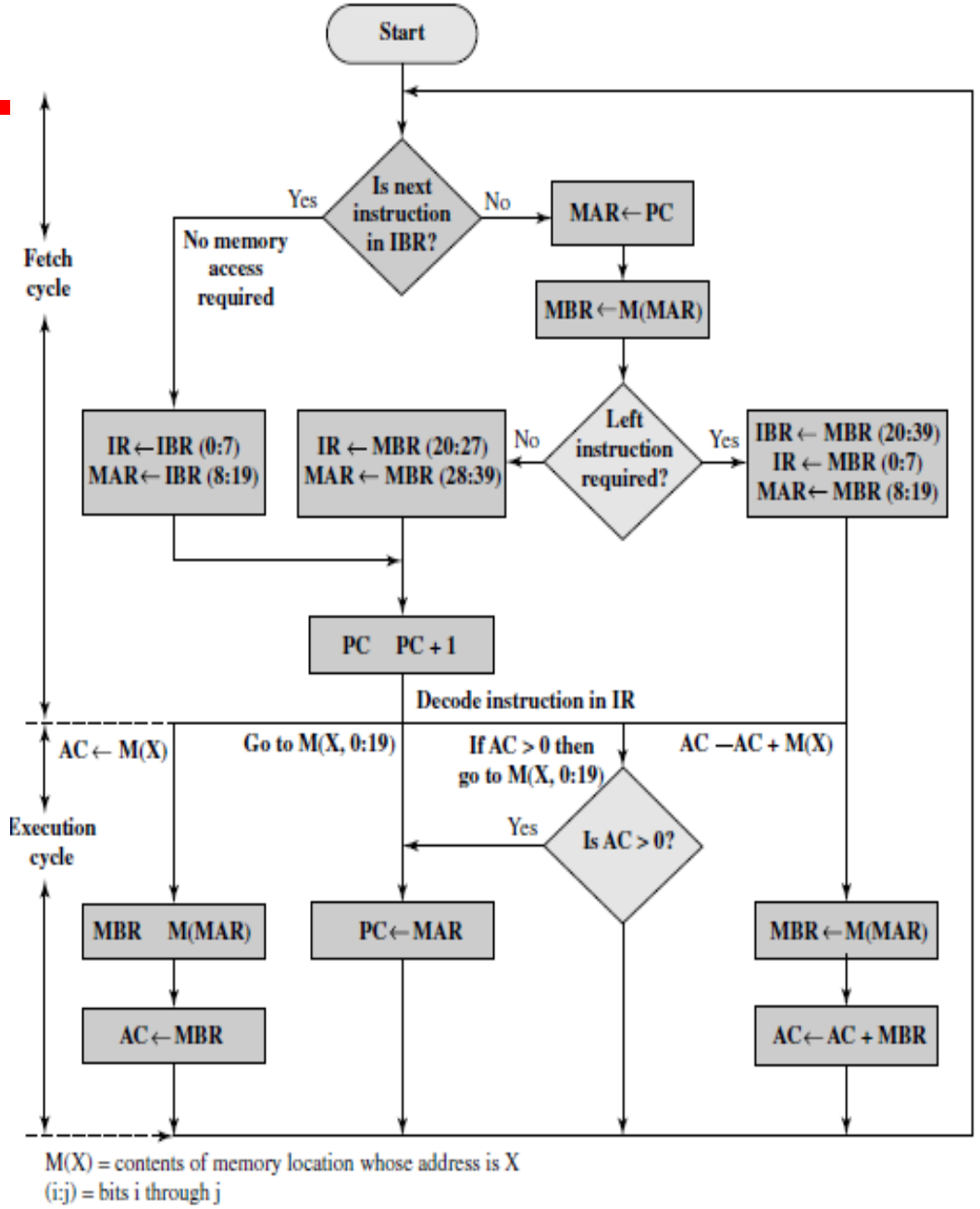
- The control unit operates the IAS by fetching instructions from memory and executing them one at a time needed, as indicated in the Figure. This figure reveals that both the control unit and

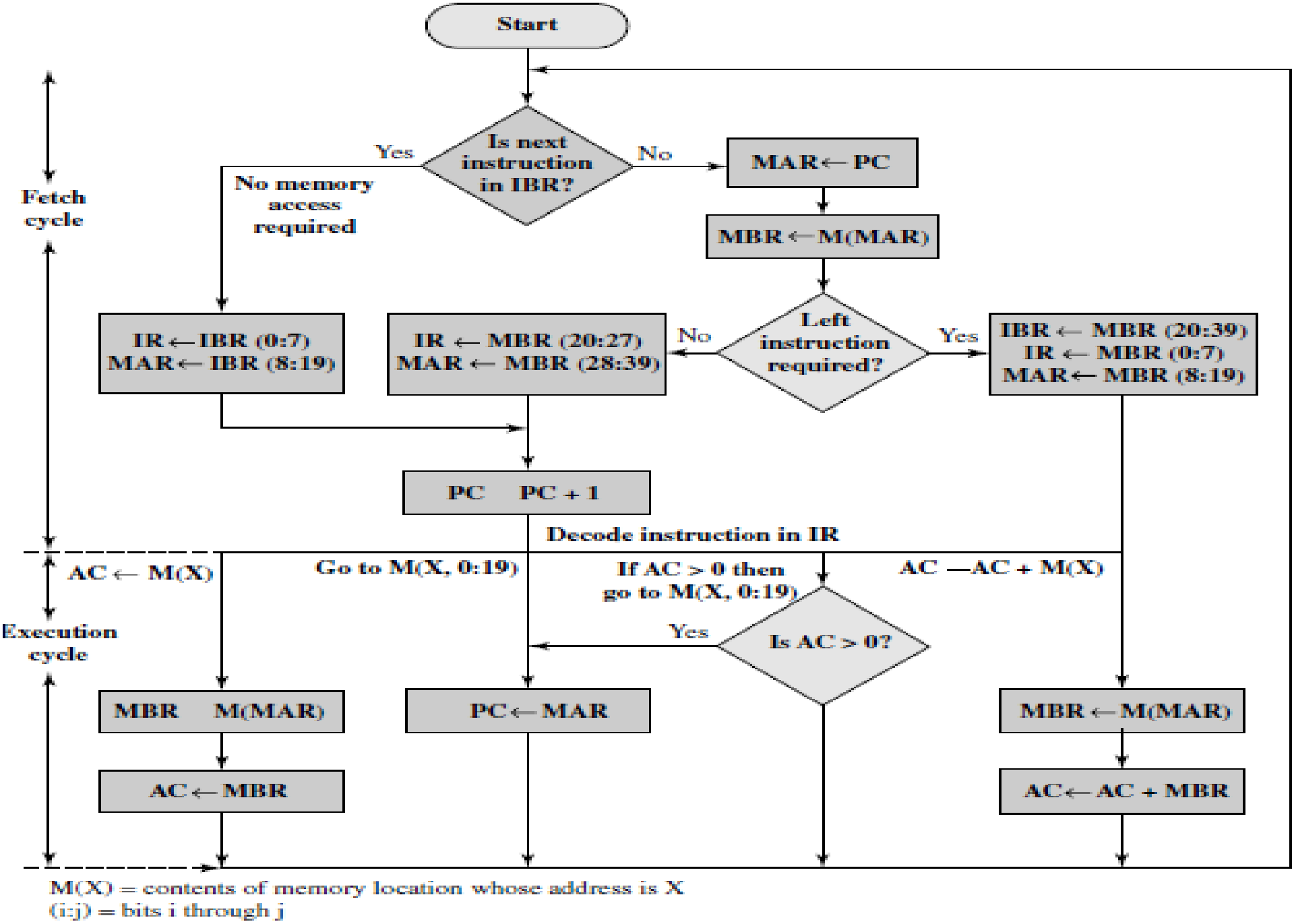


IAS - details

The IAS operates by repetitively performing an *instruction cycle*, *as shown in the* Figure. Each instruction cycle consists of two subcycles. During the ***fetch cycle,*** the opcode of the next instruction is loaded into the IR and the address portion is loaded into the MAR.

This instruction may be taken from the IBR, **or** it can be obtained from memory by loading a word into the MBR, and then down to the IBR, IR, and MAR




Start
Is next instruction in IBR?
Yes
No
No memory access required
MAR ← PC
MBR ← M(MAR)
Left instruction required?
No
Yes
IR ← IBR (0:7)
MAR ← IBR (8:19)
IR ← MBR (20:27)
MAR ← MBR (28:39)
IBR ← MBR (20:39)
IR ← MBR (0:7)
MAR ← MBR (8:19)
PC PC + 1
Decode instruction in IR
Fetch cycle
Execution cycle
AC ← M(X)
Go to M(X, 0:19)
If AC > 0 then go to M(X, 0:19)
AC —AC + M(X)
Is AC > 0?
Yes
MBR M(MAR)
PC ← MAR
MBR ← M(MAR)
AC ← MBR
AC ← AC + MBR
M(X) = contents of memory location whose address is X
(i:j) = bits i through j

IAS – Instruction Set

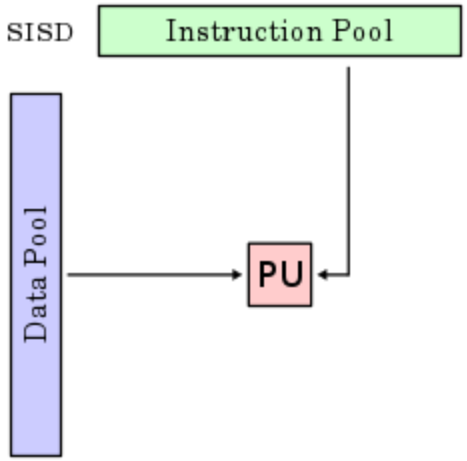
- The IAS computer had a total of 21 instructions,
- The IAS instructions can be grouped as follows:

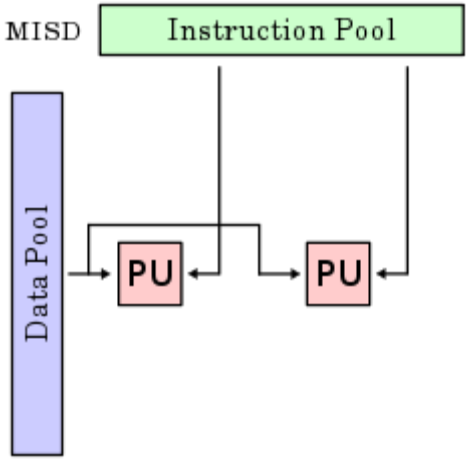
1. **Data transfer:** Move data between memory and ALU registers or between two ALU registers.
2. **Unconditional branch:** Normally, the control unit executes instructions in sequence from memory. This sequence can be changed by a branch instruction, which facilitates repetitive operations.
3. **Conditional branch:** The branch can be made dependent on a condition, thus allowing decision points.
4. **Arithmetic:** Operations performed by the ALU.
5. **Address modify:** Permits addresses to be computed in the ALU and then inserted into instructions stored in memory. This allows a program considerable addressing flexibility.

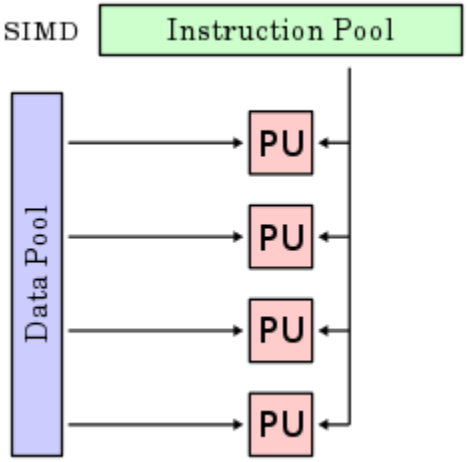
Table 2.1 The IAS Instruction Set

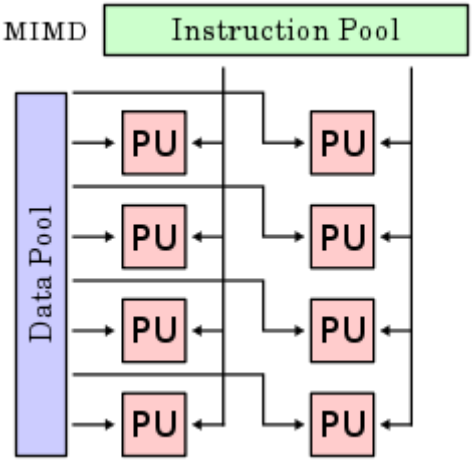
Instruction Type	Opcode	Symbolic Representation	Description
Data transfer	00001010	LOAD MQ	Transfer contents of register MQ to the accumulator AC
	00001001	LOAD MQ,M(X)	Transfer contents of memory location X to MQ
	00100001	STOR M(X)	Transfer contents of accumulator to memory location X
	00000001	LOAD M(X)	Transfer M(X) to the accumulator
	00000010	LOAD −M(X)	Transfer −M(X) to the accumulator
	00000011	LOAD |M(X)|	Transfer absolute value of M(X) to the accumulator
	00000100	LOAD −|M(X)|	Transfer −|M(X)| to the accumulator
Unconditional branch	00001101	JUMP M(X,0:19)	Take next instruction from left half of M(X)
	00001110	JUMP M(X,20:39)	Take next instruction from right half of M(X)
Conditional branch	00001111	JUMP+ M(X,0:19)	If number in the accumulator is nonnegative, take next instruction from left half of M(X)
	00010000	JUMP+ M(X,20:39)	If number in the accumulator is nonnegative, take next instruction from right half of M(X)
Arithmetic	00000101	ADD M(X)	Add M(X) to AC; put the result in AC
	00000111	ADD |M(X)|	Add |M(X)| to AC; put the result in AC
	00000110	SUB M(X)	Subtract M(X) from AC; put the result in AC
	00001000	SUB |M(X)|	Subtract |M(X)| from AC; put the remainder in AC
	00001011	MUL M(X)	Multiply M(X) by MQ; put most significant bits of result in AC, put least significant bits in MQ
	00001100	DIV M(X)	Divide AC by M(X); put the quotient in MQ and the remainder in AC
	00010100	LSH	Multiply accumulator by 2; i.e., shift left one bit position
	00010101	RSH	Divide accumulator by 2; i.e., shift right one position
Address modify	00010010	STOR M(X,8:19)	Replace left address field at M(X) by 12 rightmost bits of AC
	00010011	STOR M(X,28:39)	Replace right address field at M(X) by 12 rightmost bits of AC

Flynn's Classification









Transistors

- Replaced vacuum tubes
- Smaller
- Cheaper
- Less heat dissipation
- Solid State device
- Made from Silicon (Sand)
- Invented 1947 at Bell Labs
- William Shockley et al.

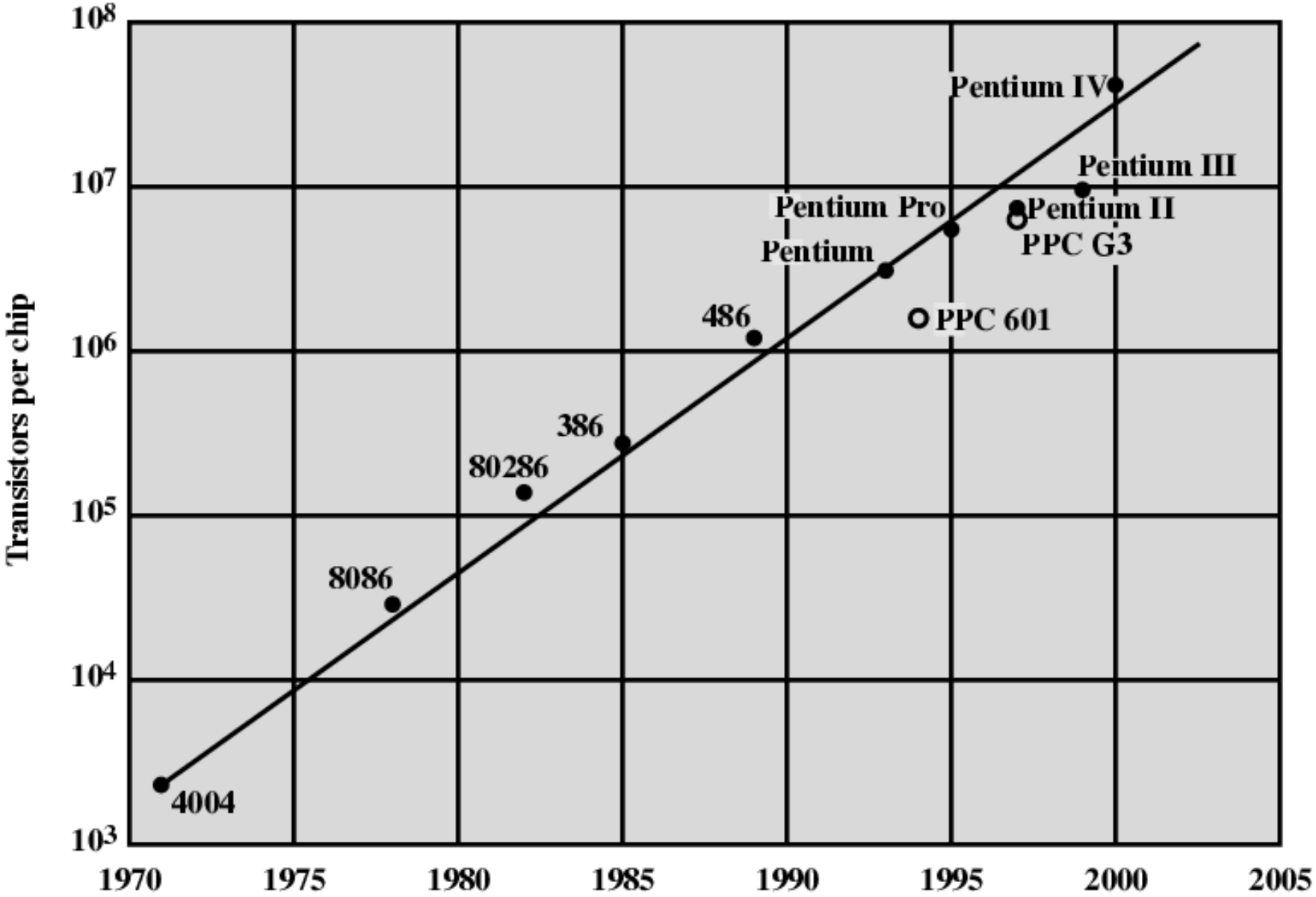
Microelectronics

- Literally - “small electronics”
- A computer is made up of gates, memory cells and interconnections
- These can be manufactured on a semiconductor
- e.g. silicon wafer

Moore's Law

- Increased density of components on chip
- Gordon Moore - cofounder of Intel
- Number of transistors on a chip will double every year
- Since 1970's development has slowed a little
  - Number of transistors doubles every 18 months
- Cost of a chip has remained almost unchanged
- Higher packing density means shorter electrical paths, giving higher performance
- Smaller size gives increased flexibility
- Reduced power and cooling requirements
- Fewer interconnections increases reliability

Growth in CPU Transistor Count



Speeding it up

- Pipelining
- On board cache
- On board L1 & L2 cache
- Branch prediction
- Data flow analysis
- Speculative execution

Performance Mismatch

- Processor speed increased
- Memory capacity increased
- Memory speed lags behind processor speed

Solutions to performance mismatch problem:

- Increase number of bits retrieved at one time
  - Make DRAM "wider" rather than "deeper"
- Change DRAM interface
  - Cache
- Reduce frequency of memory access
  - **More complex cache** and cache on chip
- Increase interconnection bandwidth
  - High speed buses
  - Hierarchy of buses

Pentium Evolution (1)

- 8080
  - first general purpose microprocessor
  - 8 bit data path
  - Used in first personal computer – Altair
- 8086
  - much more powerful
  - 16 bit
  - instruction cache, prefetch few instructions
  - 8088 (8 bit external bus) used in first IBM PC
- 80286
  - 16 Mbyte memory addressable
  - up from 1Mb
- 80386
  - 32 bit
  - Support for multitasking

Pentium Evolution (2)

- 80486
  - —sophisticated powerful cache and instruction pipelining
  - —built in maths co-processor
- Pentium
  - —Superscalar
  - —Multiple instructions executed in parallel
- Pentium Pro
  - —Increased superscalar organization
  - —Aggressive register renaming
  - —branch prediction
  - —data flow analysis
  - —speculative execution

Pentium Evolution (3)

- Pentium II
  - —MMX technology
  - —graphics, video & audio processing
- Pentium III
  - —Additional floating point instructions for 3D graphics
- Pentium 4
  - —Note Arabic rather than Roman numerals
  - —Further floating point and multimedia enhancements
- Itanium
  - —64 bit
  - —see chapter 15
- See Intel web pages for detailed information on processors

Table 2.6 Evolution of Intel Microprocessors

(a) 1970s Processors

	4004	8008	8080	8086	8088
Introduced	1971	1972	1974	1978	1979
Clock speeds	108 kHz	108 kHz	2 MHz	5 MHz, 8 MHz, 10 MHz	5 MHz, 8 MHz
Bus width	4 bits	8 bits	8 bits	16 bits	8 bits
Number of transistors	2,300	3,500	6,000	29,000	29,000
Feature size (μm)	10		6	3	6
Addressable memory	640 Bytes	16 KB	64 KB	1 MB	1 MB

(b) 1980s Processors

	80286	386TM DX	386TM SX	486TM DX CPU
Introduced	1982	1985	1988	1989
Clock speeds	6 MHz–12.5 MHz	16 MHz–33 MHz	16 MHz–33 MHz	25 MHz–50 MHz
Bus width	16 bits	32 bits	16 bits	32 bits
Number of transistors	134,000	275,000	275,000	1.2 million
Feature size (μm)	1.5	1	1	0.8–1
Addressable memory	16 MB	4 GB	16 MB	4 GB
Virtual memory	1 GB	64 TB	64 TB	64 TB
Cache	—	—	—	8 kB