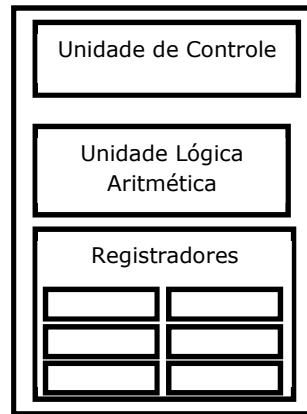


CPU (Central Processing Unit)

O Processador (CPU) é organizado em três partes:

1. **Unidade lógica aritmética (ALU):** Executa as micro-operações (soma, subtração, operações lógicas - *and*, *ou*, etc.) necessárias para executar as instruções;
2. **Registradores:** Armazena dados intermediários durante a execução de instruções;
3. **Unidade de controle:** supervisiona a transferência de informações entre os registros e instrui a ULA sobre qual operação a executar através da geração de sinais de controle.



- Execução de um programa
 - Busca a instrução na memória
 - Decodifica a instrução
 - Executa a instrução

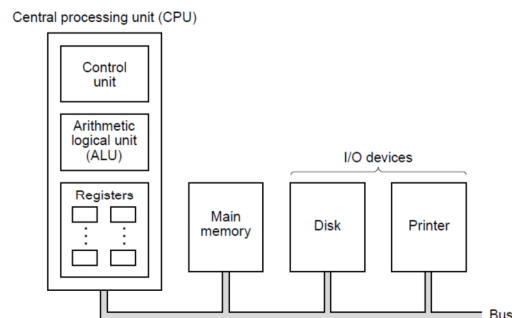


Figure 2-1. The organization of a simple computer with one CPU and two I/O devices.

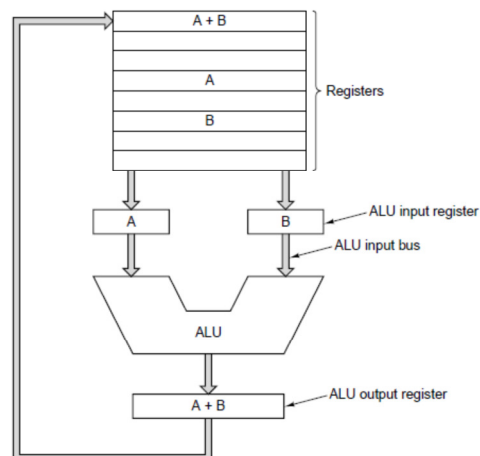


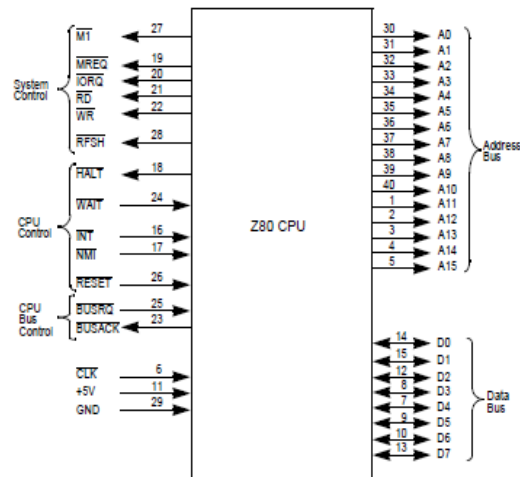
Figure 2-2. The data path of a typical von Neumann machine.

- A CPU executa um programa, que é uma sequência de instruções armazenadas.
 - Cada modelo de processador tem um conjunto de instruções, que ele executa.
- A CPU executa o programa de processamento de cada pedaço de dados conforme a orientação do programa e do conjunto de instruções.

Enquanto a CPU está executando uma etapa do programa, as demais instruções e os dados são armazenados nas proximidades de uma memória especial chamada cache.

CPU comerciais

CPU modelo: **Z80** – Fabricante: Zilog



Linhas de dados: 8 bits - D0 a D7

Linhas de endereço: 16 bits – A0 a A15

Acesso a uma memória física (principal) de até: 65.536 bytes – 64 KB

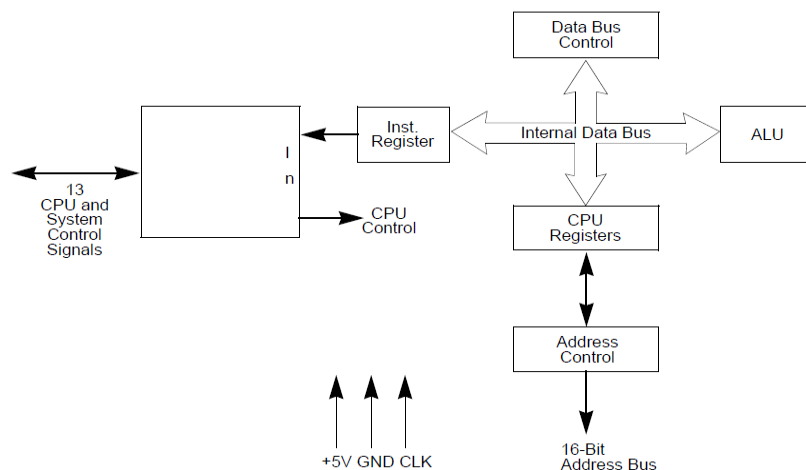
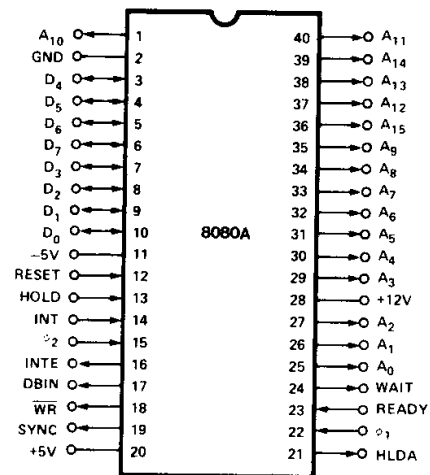


Figure 1. Z80 CPU Block Diagram

CPU modelo: **8080** – Fabricante: Intel

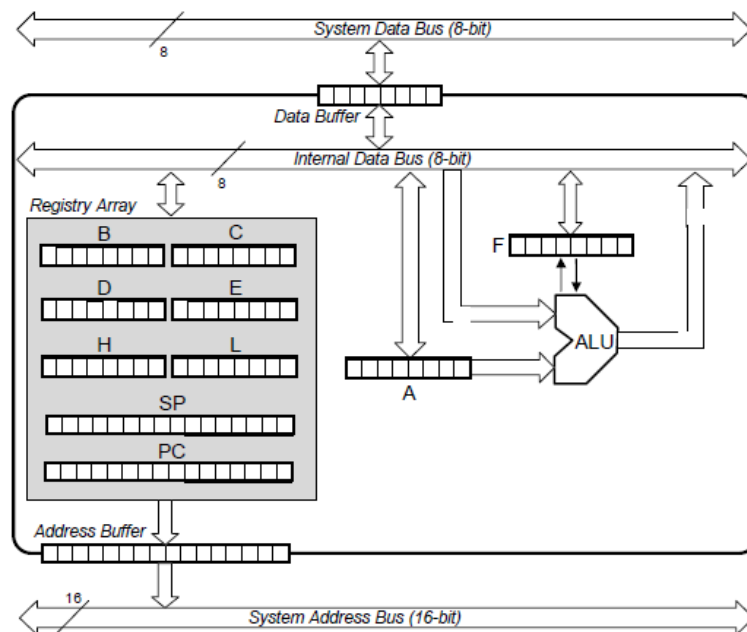


Linhas de dados: 8 bits - D0 a D7

Linhas de endereço: 16 bits – A0 a A15

Acesso a uma memória física (principal) de até: 65.536 bytes – 64 KB

Intel 8080 CPU block diagram



CPU modelo: **80286** – Fabricante: Intel

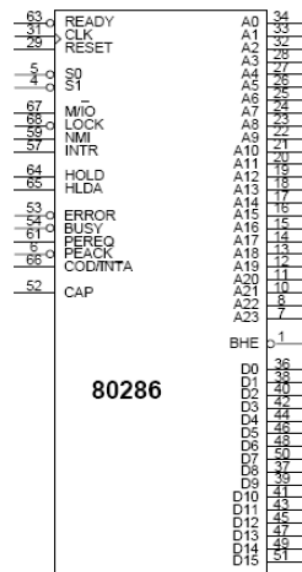


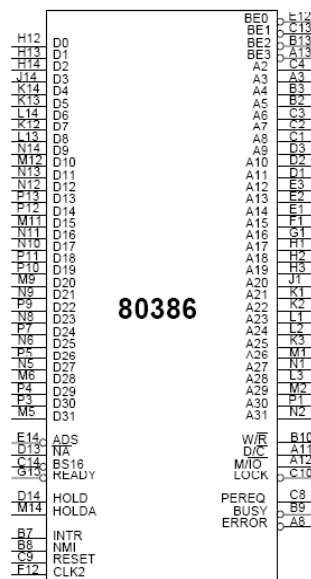
Figura 3.1. Pinagem do 80286.

Linhas de dados: 16 bits - D0 a D15

Linhas de endereço: 24 bits – A0 a A23

Acesso a uma memória física (principal) de até: 16.777.216 bytes – 16 MB

CPU modelo: **80386** – Fabricante: Intel



Pinagem do 80386, com encapsulamento 132 - PGA

Linhas de dados: 32 bits - D0 a D31

Linhas de endereço: 32 bits – A0 a A31

Acesso a uma memória física (principal) de até: 4.294.967.296 bytes – 4 GB

- O que uma CPU pode fazer está definido na *Instruction Set Architecture* (ISA)
- Uma linguagem deve ser desenvolvida para permitir ao programador dizer à CPU o que fazer (linguagem de montagem (*assembly*))
- Um programa que traduza a linguagem de montagem para bits que o computador entenda então é chamado de *assembler*
- Imaginemos uma CPU (fig. 2-2) que tem a habilidade de executar 4 operações de 2 números:
 - Adição, subtração, multiplicação e divisão.
 - Estas operações receberão respectivamente os mnemônicos:
 - *ADD, SUB, MUL* e *DIV*
 - Esta CPU tem 4 posições de memória que pode armazenar estes números:
 - R1, R2, R3 e R4
 - Imaginemos que R1 = 4, R2 = 7, R3 = 5 e R4 = 1
 - Para somar 4 + 7 teríamos então:
 - *ADD R1, R2*
 - Para subtrair 5 - 1 teríamos:
 - *SUB R3, R4*
 - Estas instruções são úteis, mas...
 - O computador não é capaz de entender, é preciso convertê-las em binário.

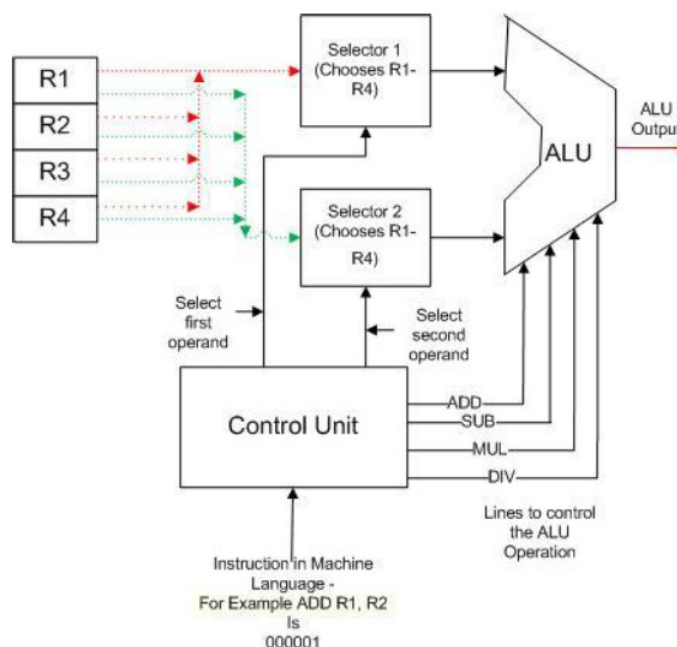


Figure 2-2: Simple CPU

- Primeiro cada operação *ADD, SUB, MUL, DIV* será traduzida para um número de 2 bits:
 - *ADD* = 00₂, *SUB* = 01₂, *MUL* = 10₂, *DIV* = 11₂

- A memória será acessada por um local, e este local terá um endereço.
- Temos 4 localizações de memória:
 - $R1, R2, R3$, e $R4$
 - O endereço de cada localização na base 2 será:
 - $R1 = 00_2, R2 = 01_2, R3 = 10_2, R4 = 11_2$
- A linguagem utilizada no computador, chamada linguagem de máquina, é uma série de 1s e 0s que representam a operação e os dois locais de memória a ser usada para recuperar os valores.
- Cada instrução será formatada com um código de operação de 2 bits (opcode), a primeira localização de memória a ser usada e a segunda localização de memória a ser usada.

OpCode	First Register	Second Register
--------	----------------	-----------------

- A instrução "**ADD R1, R2**" é traduzida então em código de máquina para o binário 000001

OpCode	First Register	Second Register
00	00	01

- e "**SUB R3, R4**" é traduzido para 011011

OpCode	First Register	Second Register
01	10	11

Exemplos

Intel 8080 instruction set architecture

Instruction code formats:

– One byte instructions:

i_7	i_6	i_5	i_4	i_3	i_2	i_1	i_0
-------	-------	-------	-------	-------	-------	-------	-------

All bits used to encode instruction, no operands.

i_7	i_6	i_5	i_4	i_3	r	r	r
-------	-------	-------	-------	-------	-----	-----	-----

One operand in 8-bit internal register A to L.

i_7	i_6	r	r	r	i_2	i_1	i_0
-------	-------	-----	-----	-----	-------	-------	-------

One operand in 8-bit internal register A to L.

i_7	i_6	$r1$	$r1$	$r1$	$r2$	$r2$	$r2$
-------	-------	------	------	------	------	------	------

Two operands in two 8-bit internal registers A to L.

i_7	i_6	p	p	i_3	i_2	i_1	i_0
-------	-------	-----	-----	-------	-------	-------	-------

One operand in 16-bit register pair BC, DE, HL or in SP.

– Two byte instructions:

i_7	i_6	i_5	i_4	i_3	i_2	i_1	i_0
d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0

All bits of the first byte used to encode instruction, the second byte is the *immediate* operand (argument).

i_7	i_6	r	r	r	i_2	i_1	i_0
d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0

First operand in 8-bit internal register A to L, the second operand is *immediate*.

i_7	i_6	i_5	i_4	i_3	i_2	i_1	i_0
p_7	p_6	p_5	p_4	p_3	p_2	p_1	p_0

All bits of the first byte used to encode instruction, the second byte is the 8-bit address of I/O port.

– Three byte instructions with immediate memory addressing:

i_7	i_6	p	p	i_3	i_2	i_1	i_0
l_7	l_6	l_5	l_4	l_3	l_2	l_1	l_0
h_7	h_6	h_5	h_4	h_3	h_2	h_1	h_0

First operand in 16-bit register pair BC, DE, HL or in SP, the second byte is *lower* part of 16-bit second operand, the third byte is *higher* part of the 16-bit second operand.

Instruction list:

Data transfer instructions

MOV R1, R2 (Move register)

0	1	r1	r1	r1	r2	r2	r2
---	---	----	----	----	----	----	----

 $R1 \leftarrow R2$ data from R2 is copied to R1

MOV R, M (Move from memory, address in HL)

0	1	r	r	r	1	1	0
---	---	---	---	---	---	---	---

 $R \leftarrow [HL]$ data from memory (address in HL) copied to R

MOV M, R (Move to memory, address in HL)

0	1	1	1	0	r	r	r
---	---	---	---	---	---	---	---

 $[HL] \leftarrow R$ data from R copied to memory (address in HL)

MVI R, data8 (Move to register immediate)

0	0	r	r	r	1	1	0
d ₇	d ₆	d ₅	d ₄	d ₃	d ₂	d ₁	d ₀

 $R \leftarrow \text{data8}$ 1 byte (next to instruction) copied to R

MVI M, data8 (Move to memory immediate)

0	0	1	1	0	1	1	0
d ₇	d ₆	d ₅	d ₄	d ₃	d ₂	d ₁	d ₀

 $[HL] \leftarrow \text{data8}$ 1 byte copied to memory (address in HL)

Table 2-1. Processor Performance Over Time and Other Key Features of the Intel Architecture

Intel Processor	Date of Product Introduction	Performance in MIPS ¹	Max. CPU Frequency at Introduction	No. of Transistors on the Die	Main CPU Register Size ²	Extern. Data Bus Size ²	Max. Extern. Addr. Space	Caches in CPU Package ³
8086	1978	0.8	8 MHz	29 K	16	16	1 MB	None
Intel 286	1982	2.7	12.5 MHz	134 K	16	16	16 MB	Note 3
Intel386™ DX	1985	6.0	20 MHz	275 K	32	32	4 GB	Note 3
Intel486™ DX	1989	20	25 MHz	1.2 M	32	32	4 GB	8KB L1
Pentium®	1993	100	60 MHz	3.1 M	32	64	4 GB	16KB L1
Pentium Pro	1995	440	200 MHz	5.5 M	32	64	64 GB	16KB L1; 256KB or 512KB L2

Structured Computer Organization, Fifth Edition, Andrew S. Tanenbaum

Digital circuit projects - Charles W. Kann III

Z80 Datasheet - Zilog

8080A Datasheet – Intel

80286 Datasheet – Intel

80386 Datasheet - Intel