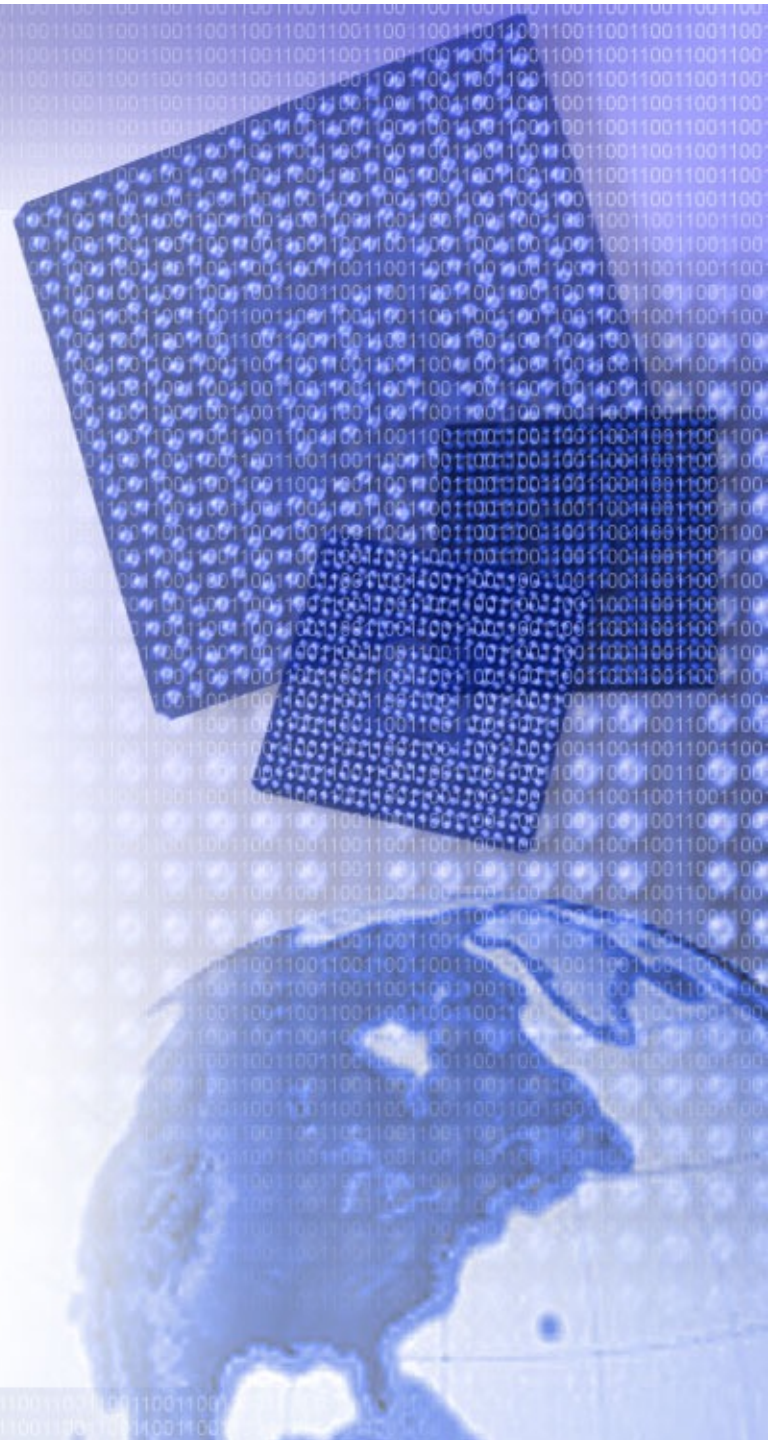
SEL0629 - Aplicação de Microprocessadores I

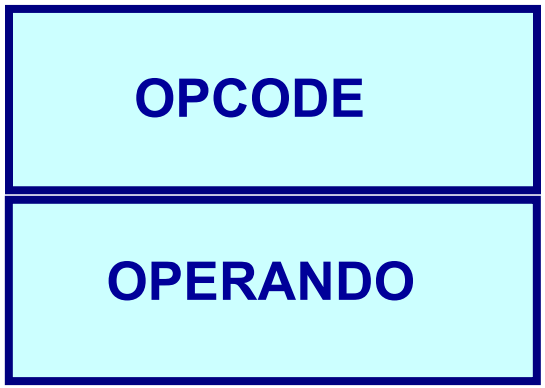
Aula 1
Set de Instruções e Arquiteturas de Sistemas Embarcados

Marcelo Andrade da Costa Vieira



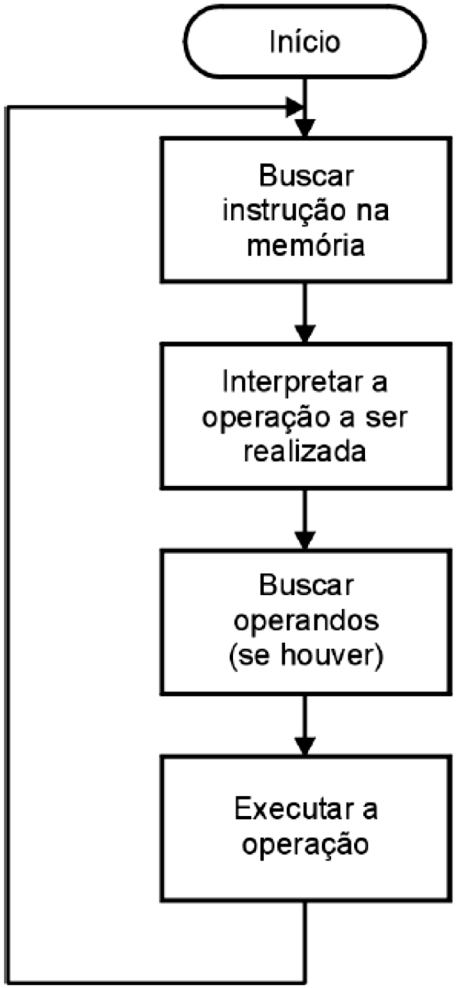
Classificação: Conjunto de Instruções

CISC - *Complex Instruction Set Computers*



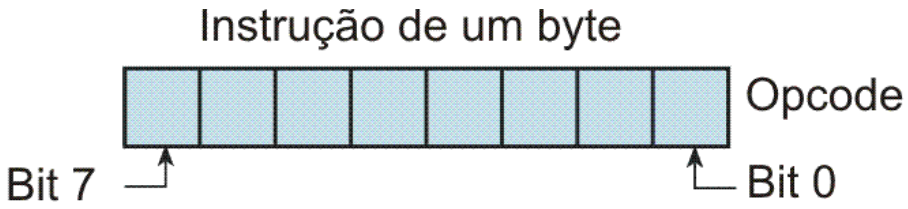
- *Opcode e Operando são armazenados em diferentes posições na memória ROM;*
- *Suporta um conjunto maior de instruções;*
- *Processamento mais lento (instruções mais complexas que gastam mais de um ciclo de máquina);*
- *Em geral, os programas são menores em no. de instruções, mas ocupam mais espaço na memória de programa*

Ciclo de Instrução - CISC

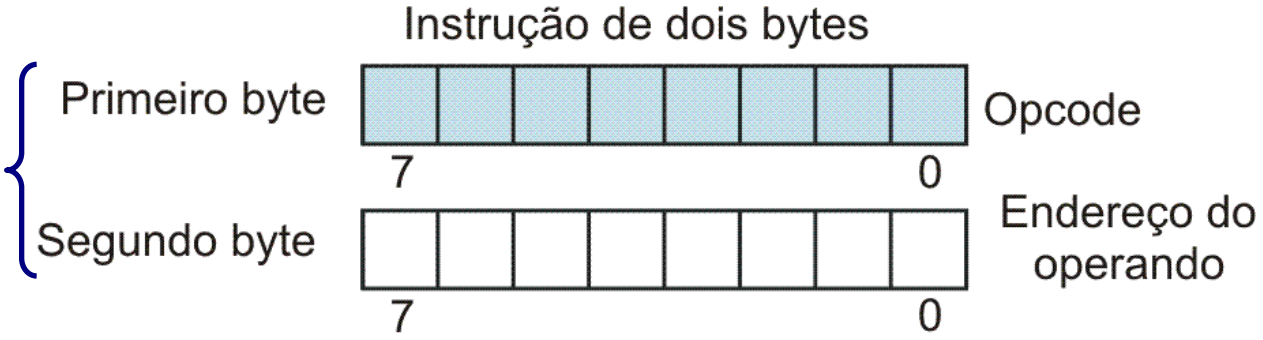


Exemplos de Instruções CISC

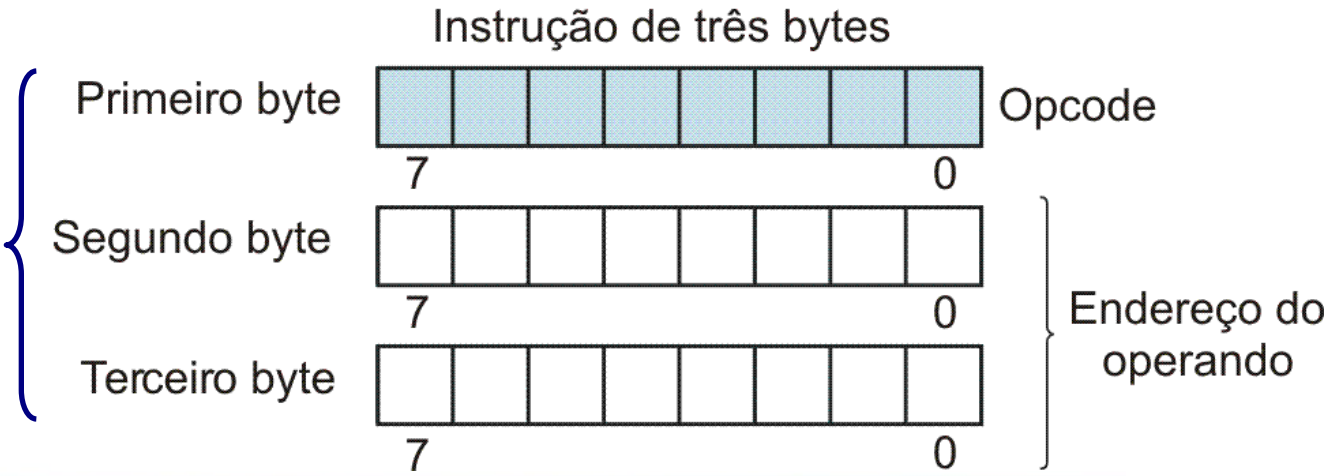
CLR A



MOV A,30h



LJMP 3FB2h



Algumas instruções do 8051 (CISC)

Table 1. AT89 Instruction Set Summary[1]

Mnemonic		Description	Byte	Oscillator Period
ARITHMETIC OPERATIONS				
ADD	A,R_n	Add register to Accumulator	1	12
ADD	A,direct	Add direct byte to Accumulator	2	12
ADD	A,@R_i	Add indirect RAM to Accumulator	1	12
ADD	A,#data	Add immediate data to Accumulator	2	12
ADDC	A,R_n	Add register to Accumulator with Carry	1	12
ADDC	A,direct	Add direct byte to Accumulator with Carry	2	12
ADDC	A,@R_i	Add indirect RAM to Accumulator with Carry	1	12
ADDC	A,#data	Add immediate data to Acc with Carry	2	12
SUBB	A,R_n	Subtract Register from Acc with borrow	1	12
SUBB	A,direct	Subtract direct byte from Acc with borrow	2	12
SUBB	A,@R_i	Subtract indirect RAM from ACC with borrow	1	12
SUBB	A,#data	Subtract immediate data from Acc with borrow	2	12
INC	A	Increment Accumulator	1	12
INC	R_n	Increment register	1	12
INC	direct	Increment direct byte	2	12
INC	@R_i	Increment direct RAM	1	12
DEC	A	Decrement Accumulator	1	12
DEC	R_n	Decrement Register	1	12
DEC	direct	Decrement direct byte	2	12
DEC	@R_i	Decrement indirect RAM	1	12
INC	DPTR	Increment Data Pointer	1	24
MUL	AB	Multiply A & B	1	48
DIV	AB	Divide A by B	1	48
DA	A	Decimal Adjust Accumulator	1	12

Computadores CISC

Exemplo de um Programa Assembly do 8051

Memória

00	E5
01	30
02	B4
03	00
04	FB
05	80
06	FE

Endereço Conteúdo

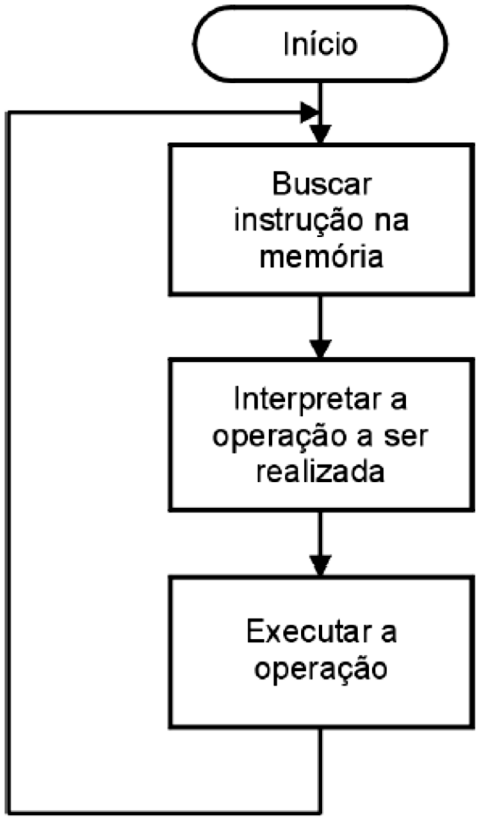
Addr	Opcodes	ASC	Label	Disassembly
0000	E5 30	å0	LOOP	MOV A,30h
0002	B4 00 FB	´ù		CJNE A,#00h,LOOP
0005	80 FE	€þ	AQUI	SJMP AQUI

Classificação: Conjunto de Instruções

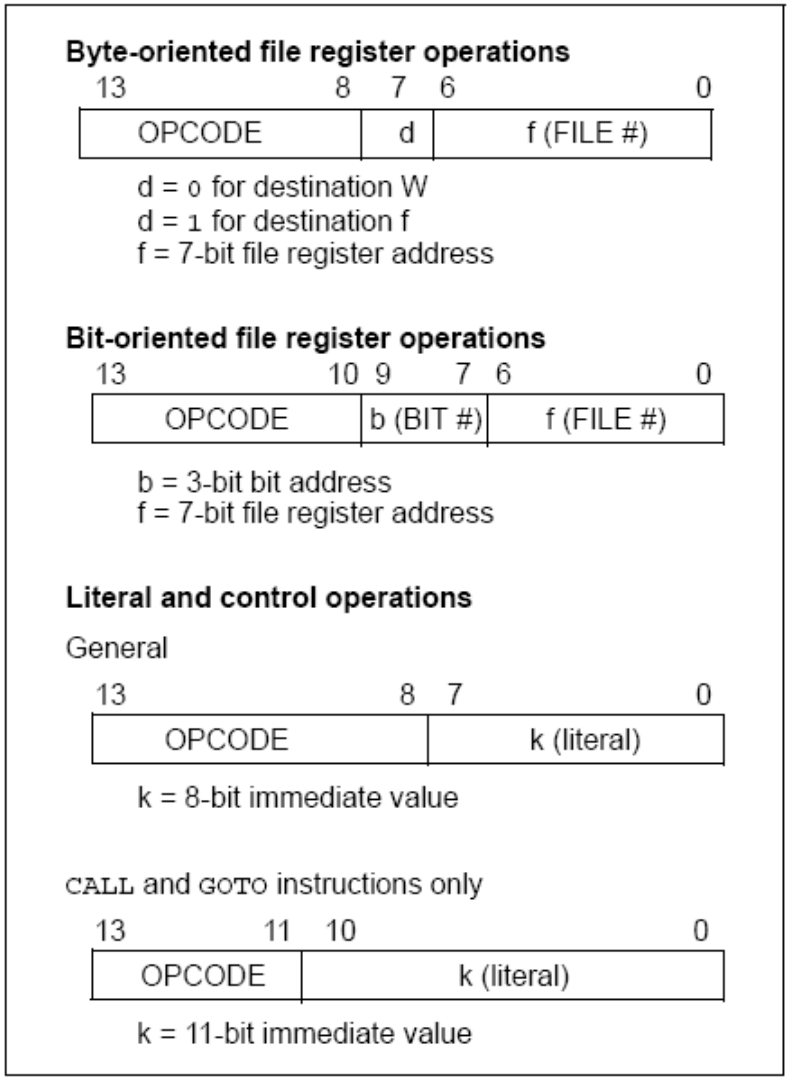
RISC - *Reduced Instruction Set Computers*

- *Opcode e Operando são armazenados na mesma posição na memória ROM;*
- *Implementa quantidade limitada de instruções ⇨ otimizadas ⇨ maior rapidez*
- *Gastam em sua maioria apenas um ciclo de máquina;*
- *Todas as instruções ocupam o mesmo tamanho na memória*
- *Em geral, os programas são maiores em no. de instruções, mas ocupam menos espaço na memória.*

Ciclo de Instrução - RISC



Exemplos de Instruções RISC



MOVF STATUS, W

BCF STATUS, RP0

MOVLW B‘00011100’

CALL SUBROTINA

Instruções do PIC16F877

TABLE 15-2: PIC16F87XA INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb			LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	-	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	f	Move W to f	1	00	0000	lfff	ffff		
NOP	-	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$,$\overline{PD}$	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	-	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	-	Go into Standby mode	1	00	0000	0110	0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

CISC x RISC

CISC:

Mais instruções disponíveis = programa mais simples.

Instruções podem ocupar espaços diferentes na memória de programa (Opcode + operando)

Acabam tendo durações diferentes;

Quantidade de instruções num programa geralmente é menor, mas o programa final acaba ocupando mais espaço na memória de programa;

RISC:

Menos instruções disponíveis = programas mais complexos.

Cada instrução ocupa o mesmo espaço na memória de programa (Opcode + operando);

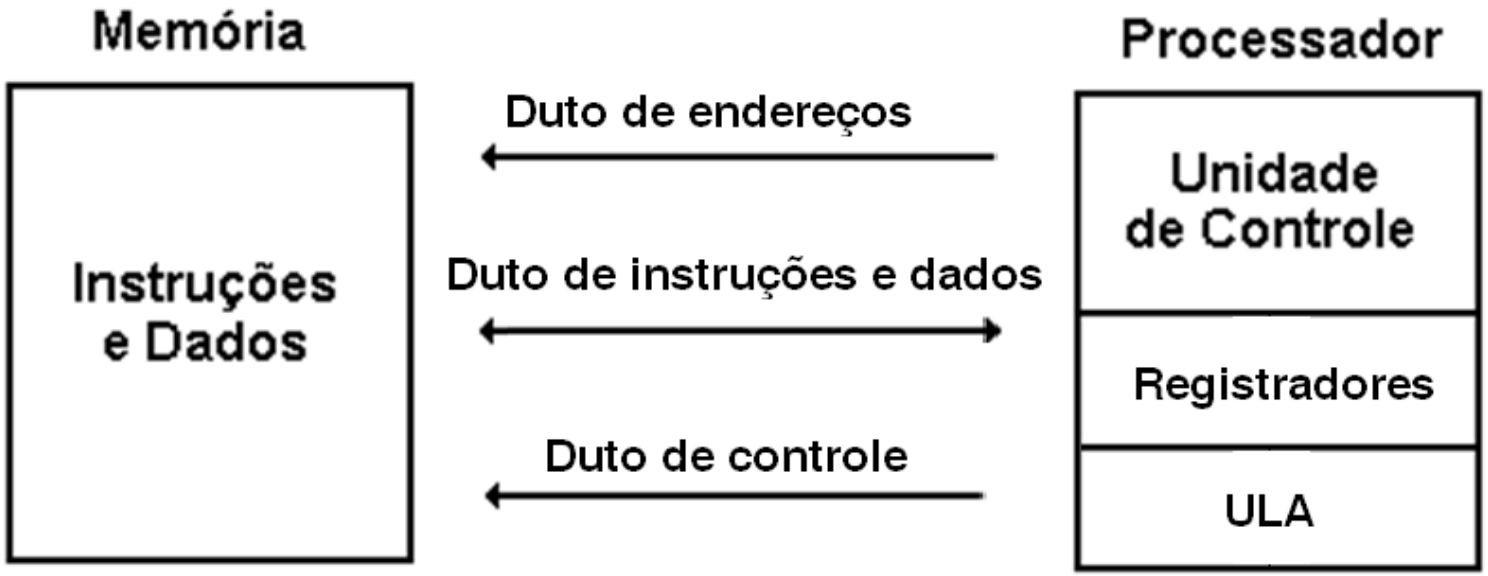
Todas tem a mesma duração (exceto as de “salto”);

Quantidade de instruções num programa geralmente é maior, mas o programa final acaba ocupando menos espaço na memória de programa;

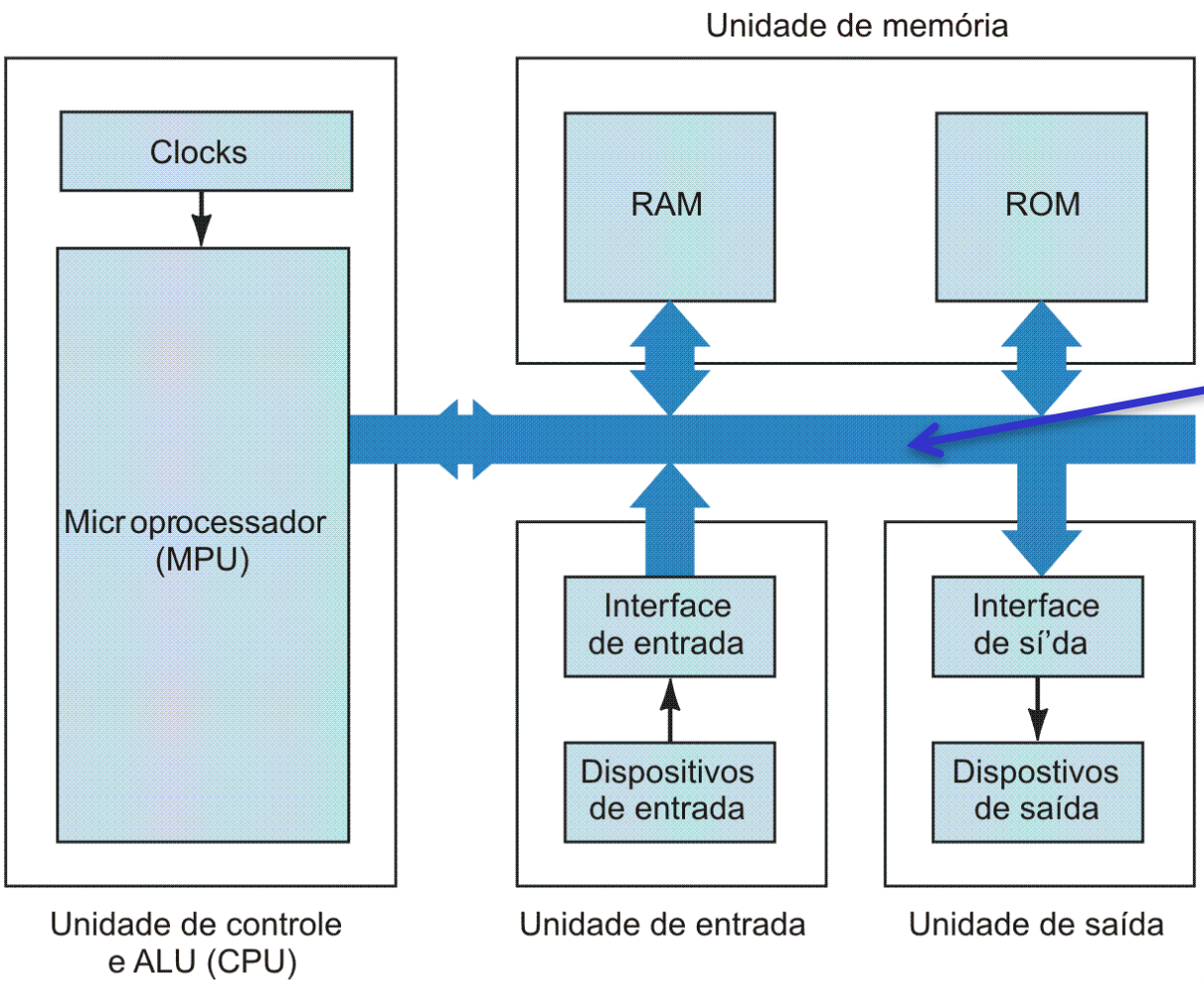
Modelos de Arquiteturas para Computadores

Arquitetura de Von Neumann
X
Arquitetura Harvard

Arquitetura Von Neumann

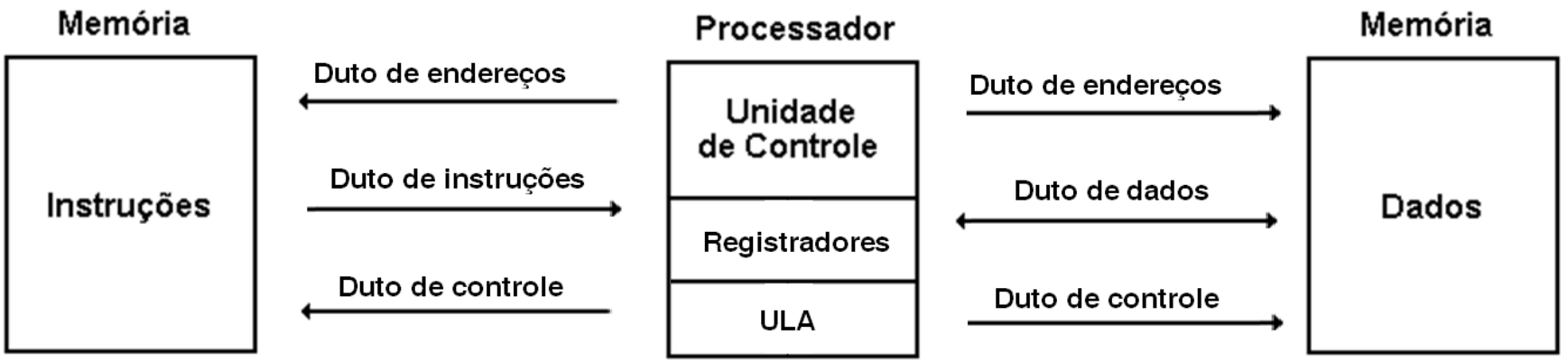


Microcontrolador Intel 8051
Arquitetura Von Neumann

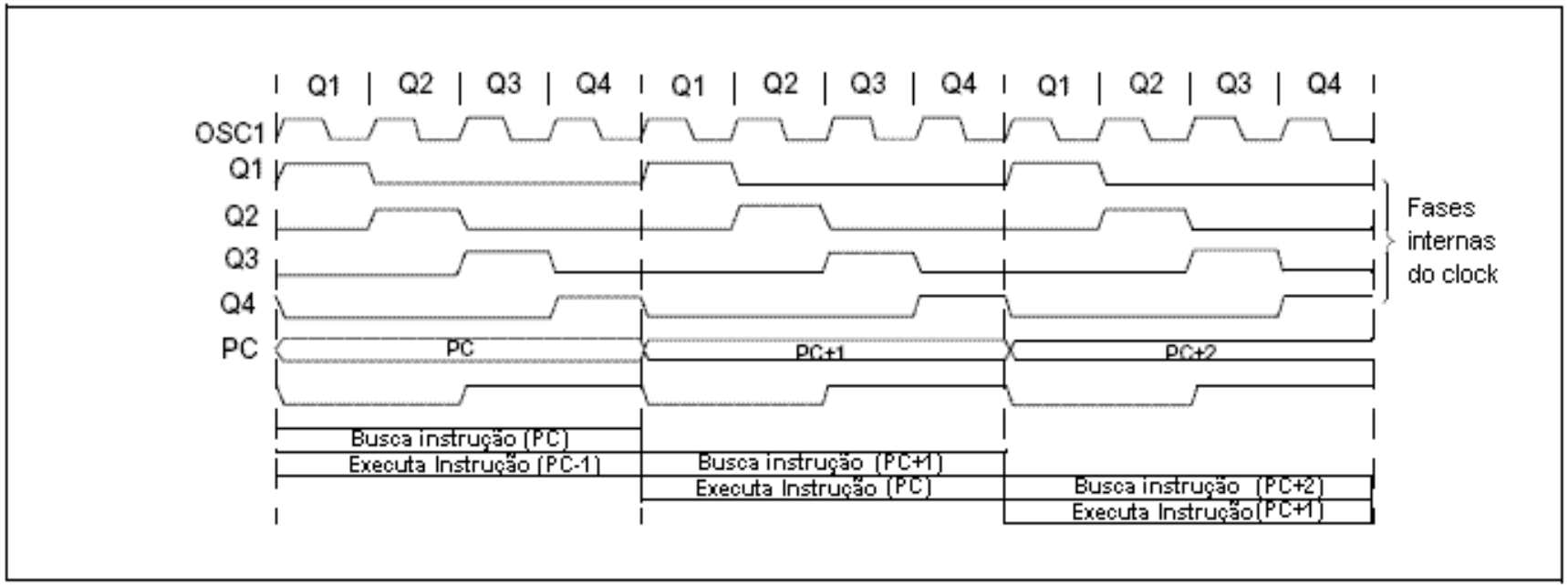


Apesar de duas memórias, elas compartilham o mesmo barramento

Arquitetura Harvard



Pipelining de 2 estágios: Exemplo do µC PIC

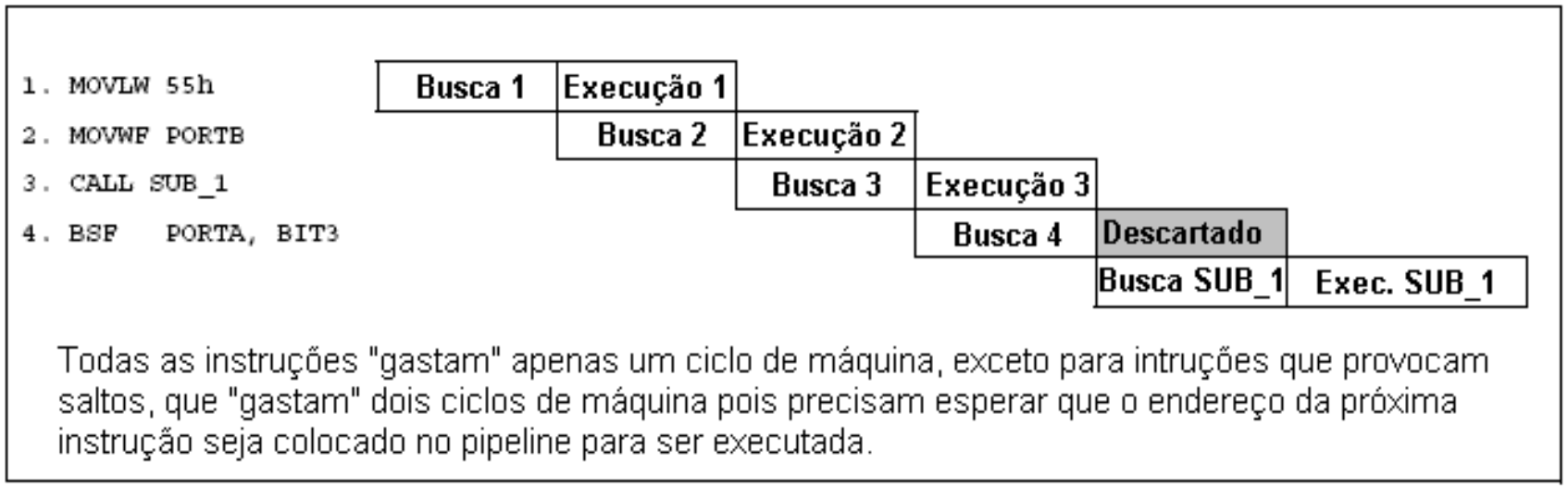


Ciclo de Busca = 4T; Ciclo de Execução = 4T;

Ciclo de máquina = (4T+4T)/2 = 4T;

Ciclo de máquina = Fosc/4;

Pipelining de 2 estágios: Exemplo do µC PIC



Todas as instruções "gastam" apenas um ciclo de máquina, exceto para intruções que provocam saltos, que "gastam" dois ciclos de máquina pois precisam esperar que o endereço da próxima instrução seja colocado no pipeline para ser executada.

A maioria das instruções deve gastar 1 ciclo de máquina (CM);

A primeira instrução sempre gasta 2 CM;

As instruções de “salto” também gastam 2 CM.

Von Neumann X Harvard

Arquitetura Von Neumann:

Arquitetura mais simples;

Menos eficiente pois não permite acesso simultâneo às memórias – barramento compartilhado;

Pode ser CISC ou RISC;

Exemplo:

4004 – 46 instruções

8080 – 78 instruções

8051 – 111 instruções

Z80 – Mais de 500 instruções

Von Neumann X Harvard

Arquitetura Harvard:

Arquitetura mais complexa;

Mais eficiente, pois permite acesso simultâneo às memórias;

Set de instruções deve ser RISC;

Permite o *Pipelining* de 2 ou mais estágios.

Exemplo:

Intel 8086, 8088

Microchip PIC16 – 35 instruções

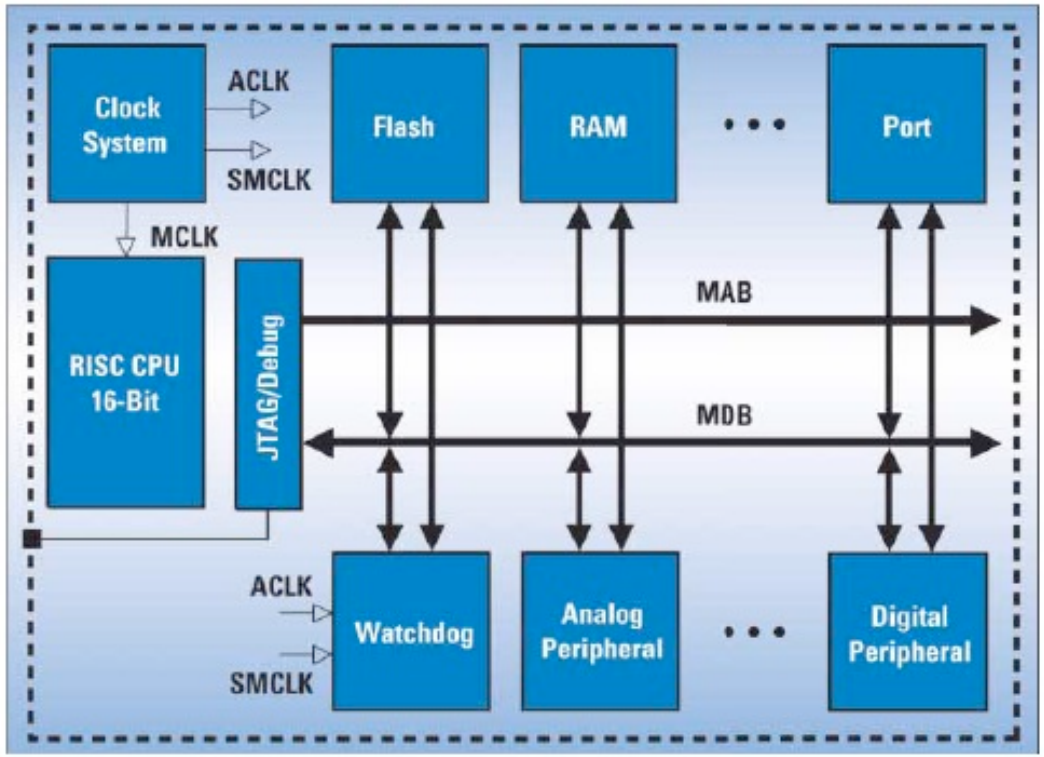
Microchip PIC18 – 75 instruções

Arquitetura Von Neummann com Set de Instruções RISC

Texas MSP430:

Arquitetura Von Neumann;

Intruções RISC de 16 bits;



FIM