

Modos de Endereçamento

8086 INSTRUCTION SET

OPCODE	DESCRIPTION		JNAE	label	Jump if not above or equal	PUSHF	Push flags onto stack
AAA		ASCII adjust addition	JNB	label	Jump if not below	RCL	dt,cnt Rotate left through carry
AAD		ASCII adjust division	JNBE	label	Jump if below or equal	RCR	dt,cnt Rotate right through carry
AAM		ASCII adjust multiply	JNC	label	Jump if no carry	REP	Repeat string operation
AAS		ASCII adjust subtraction	JNE	label	Jump if not equal	REPE	Repeat while equal
ADC	dt,sc	Add with carry	JNG	label	Jump if not greater	REPZ	Repeat while zero
ADD	dt,sc	Add	JNGE	label	Jump if not greater or equal	REPNE	Repeat while not equal
AND	dt,sc	Logical AND	JNL	label	Jump if not less	REPNZ	Repeat while not zero
CALL	proc	Call a procedure	JNLE	label	Jump if not less or equal	RET	[pop] Return from procedure
CBW		Convert byte to word	JNZ	label	Jump if not zero	ROL	dt,cnt Rotate left
CLC		Clear carry flag	JNO	label	Jump if not overflow	ROR	dt,cnt Rotate right
CDL		Clear direction flag	JNP	label	Jump if not parity	SAHF	Store AH into flags
CLI		Clear interrupt flag	JNS	label	Jump if not sign	SAL	dt,cnt Shift arithmetic left
CMC		Complement carry flag	JO	label	Jump if overflow	SHL	dt,cnt Shift logical left
CMP	dt,sc	Compare	JPO	label	Jump if parity odd	SAR	dt,cnt Shift arithmetic right
CMPS	[dt,sc]	Compare string	JP	label	Jump if parity	SBB	dt,sc Subtract with borrow
CMPSB	"	" bytes	JPE	label	Jump if parity even	SCAS	[dt] Scan string
CMPSW	"	" words	JS	label	Jump if sign	SCASB	" " byte
CWD		Convert word to double word	JZ	label	Jump if zero	SCASW	" " word
DAA		Decimal adjust addition	LAHF		Load AH from flags	SHR	dt,cnt Shift logical right
DAS		Decimal adjust subtraction	LDS	dt,sc	Load pointer using DS	STC	Set carry flag
DEC	dt	Decrement	LEA	dt,sc	Load effective address	STD	Set direction flag
DIV	sc	Unsigned divide	LES	dt,sc	Load pointer using ES	STI	Set interrupt flag
ESC	code,sc	Escape	LOCK		Lock bus	STOS	[dt] Store string
HLT		Halt	LODS	[sc]	Load string	STOSB	" " byte
IDIV	sc	Integer divide	LODSB	"	" bytes	STOSW	" " word
IMUL	sc	Integer multiply	LODSW	"	" words	SUB	dt,sc Subtraction
IN	ac,port	Input from port	LOOP	label	Loop	TEST	dt,sc Test (logical AND)
INC	dt	Increment	LOOPE	label	Loop if equal	WAIT	Wait for 8087
INT	type	Interrupt	LOOPZ	label	Loop if zero	XCHG	dt,sc Exchange
INTO		Interrupt if overflow	LOOPNE	label	Loop if not equal	XLAT	table Translate
IRET		Return from interrupt	LOOPNZ	label	Loop if not zero	XLATB	" "
JA	label	Jump if above	MOV	dt,sc	Move	XOR	dt,sc Logical exclusive OR
JAE	label	Jump if above or equal	MOVS	[dt,sc]	Move string		
JB	label	Jump if below	MOVSB	"	" bytes		
JBE	label	Jump if below or equal	MOVSW	"	" words		
JC	label	Jump if carry	MUL	sc	Unsigned multiply		
JCXZ	label	Jump if CX is zero	NEG	dt	Negate		
JE	label	Jump if equal	NOP		No operation		
JG	label	Jump if greater	NOT	dt	Logical NOT		
JGE	label	Jump if greater or equal	OR	dt,sc	Logical OR		
JL	label	Jump if less	OUT	port,ac	output to port		
JLE	label	Jump if less or equal	POP	dt	Pop word off stack		
JMP	label	Jump	POPF		Pop flags off stack		
JNA	label	Jump if not above	PUSH	sc	Push word onto stack		

Notes:

dt - destination

sc - source

label - may be near or far address

label - near address

Notes:
dt - destination
sc - source
label - may be near or far address
label - near address

Definição

Modo de endereçamento é a forma pela qual os dados são armazenados ou recuperados de uma determinada célula de memória, em um endereço, conforme a arquitetura de computadores utilizada. Em um sistema computacional digitalizado, acessar diferentes estruturas de dados de formas diversas pode acarretar em melhor desempenho.

Registrador

Modo pelo qual um ou mais operandos estão contidos em registradores.

Add R1, R2

Adiciona o conteúdo de $R1 + R2$ em $R1$.

Imediato

Modo pelo qual um operando é uma constante e será adicionado a um registrador.

Add R1, #8

Adiciona o conteúdo 8 em R1.

Direto ou Absoluto

Modo usado no acesso de dados alocados estaticamente na memória (vetores ou matrizes fixas).

Add R1, (100101)

Adiciona o conteúdo R1 + Mem(100101) em R1.

Indireto de Memória

Modo usado para dereferenciar um ponteiro (buscar o valor contido no endereço armazenado – ponteiro para ponteiro).

Add R1, @(R4)

Adiciona o conteúdo R1 + conteúdo da memória apontada por um ponteiro de R4 em R1.

Indireto Registrador

Modo usado para calcular endereços usando memória com ponteiros.

Add R1, (R4)

Adiciona o conteúdo R1 + conteúdo da memória apontada por R4 em R1.

Deslocamento

Modo usado para acessar variáveis locais.

Add R1, 100 (R2)

Adiciona o conteúdo $R1 + \text{Mem}[100 + \text{Regs}[R2]]$ em R1.

Indexado

Modo usado para acessar elementos de um vetor (R2 é a base e R3 é o índice).

Add R1, (R2 + R3)

Adiciona o conteúdo R1 +
 $\text{Mem}[\text{Regs}[\text{R2}] + \text{Regs}[\text{R3}]]$ em R1.

Autoincremento

Modo usado para percorrer vetores em um laço.

Add R1, (R2) +

R2 é inicializado com o endereço do primeiro elemento do vetor: $R1 = R1 + \text{conteúdo de } R2$. Posteriormente, R2 é incrementado com o tamanho de sua estrutura de referência.

Autodecremento

Modo usado para percorrer vetores em um laço.

Add R1, -(R2)

R2 é inicializado com o endereço do último elemento do vetor: $R1 = R1 + \text{conteúdo de } R2$. Posteriormente, R2 é decrementado com o tamanho de sua estrutura de referência.

Escalonado

Modo usado para indexar vetores com elementos de tamanhos sem convenção.

Add R1, 10 (R2) [R3]

$R1 = R1 + \text{Mem}[100 + \text{Regs}[R2] + \text{Regs}[R3] * d]$, onde d é a dimensão da estrutura de dados.

Referências

- COSTA, F. M. ISA: **Arquitetura de Conjunto de Instruções**. Disponível em: <<http://www.inf.ufg.br/~fmc/arqcomp/ISA1.pdf>>. Acesso em: 3 nov. 2018.
- McROBERTS, M. Arduino Básico. 1 ed. São Paulo: Novatec, 2011. 456p.
- SATO, G. Y. Sistemas microprocessados. Disponível em: <http://paginapessoal.utfpr.edu.br/sato/material-didatico/sistemas-microcontrolados/material-de-sistemas-microcontrolados/gs_1212_intro_avr_v0.pdf>. Acesso em: 1º set. 2018.
- SOUZA, F. Arduino Uno. Disponível em: <<https://www.embarcados.com.br/arduino-uno/>>. Acesso em: 1º set. 2018.

Dúvidas



Obrigado...

**Bom
Fim de semana
para
todos!!!**