

Микропроцесорски системи

3. блок: Архитектура микропроцесора

др Срђан Митровић, дипл. инж.
e-mail: srdjan.mitrovic@va.mod.gov.rs

Скуп инструкција – Instruction Set

Скуп расположивих инструкција је важна карактеристика једног процесора.

Контролер се бира тако да његов скуп инструкција у потуности задовољава потребе пројектовања.

Особине од утицаја на одлуку:

- величина инструкције,
- брзина извршавања,
- расположивост инструкција и
- модови адресирања.

Величина инструкције – Instruction Size

Opcode

дефинише операцију која треба да се изврши и њене операнде.

Већи *opcode* процесора не подразумева нужно да програм заузима више меморије

Пример 1

10 линија кода ATmega16 RISC процесора захтева 20 бајта (свака инструкција се декодује у 16 битова).

10 линија кода 68030 процесора захтева 4 бајта .

Архитектуре према врсти инструкција

Stack Architecture: У овој архитектури, познатој и као *0-address format architecture*, експлицитно не постоје операнди. Они су организовани у стеку.

Accumulator Architecture: (*1-address format architecture*), један операнд се увек налази у акумулатору, где се смешта и резултат.

2-address Format Architecture: Оба операнда су задата, али се резултат смешта на локацију једног од њих (АТmega16 користи први регистар)

3-address Format Architecture: Локације операнда и резултата се експлицитно задају. Најфлексибилнија – највећа инструкција

Пример

Поређење архитектура

$$(A + B) * C$$

stack	accumulator	2-address format	3-address format
PUSH A	LOAD A	LOAD R1, A	LOAD R1, A
PUSH B	ADD B	LOAD R2, B	LOAD R2, B
ADD	MUL C	ADD R1, R2	ADD R1, R1, R2
PUSH C		LOAD R2, C	LOAD R2, C
MUL		MUL R1, R2	MUL R1, R1, R2
stack	accumulator	R1	R1

Брзина извршавања – Execution Speed

Зависи од више чиниоца:

- сложеност архитектуре,
- величина речи,
- фреквенција осцилатора.

Расположивост инструкција – Available Instructions

Инструкције се обично деле на следеће класе:

Arithmetic-Logic Instructions: ADD, SUB, MUL, ... AND, OR, XOR, ... BSET (сетуј бит), BCLR (обриши бит), и BTST (да ли је бит сетован).

Data Transfer: LD (*load*) и ST (*store*), PUSH и POP.

Program Flow: скокови, условна гранања, позиви процедура, повратак из процедуре или прекида: BNE, RET или RETI, ...

Control Instructions: утичу на рад контролера: NOP (не ради ништа), ресет, sleep, управљање дебаговањем, ...

Пример: операција померања – Shift

Померање битова у регистру се може извести:

- у леву страну
- у десну страну

Померање може бити и аритметичка и логичка операција. Разлика је у третирању MSB (*Most Significant Bit*) када се померање изводи у десну страну (дељење бројем 2). За померање у леву страну (множење бројем 2), MSB нема значај.

Аритметичко померање у лево:

4-битни контролера, комплемент двојке

број -3 је бинарно 1101

померањем на лево се добија $1010 = (-6)_{10}$

Пример: операција померања – Shift

Померање битова у регистру се може извести:

- у леву страну
- у десну страну

Померање може бити и аритметичка и логичка операција. Разлика је у третирању MSB (*Most Significant Bit*) када се померање изводи у десну страну (дељење бројем 2). За померање у леву страну (множење бројем 2), MSB нема значај.

Аритметичко померање у десно:

број $-4 = 1101$

померањем на десно се добија $1110 = (-2)_{10}$ (уз сетован MSB)

логичко померање на десно даје: $0110 = (6)_{10}$

Начини адресирања

Употреба аритметичке операције

Мора се експлицитно нагласити где се операнди налазе!

Они могу бити константе, садржај регистра или садржај меморијске локације.



immediate/literal

Операнд је константа.

Регистарско – Register

Операнд је у регистру (или резултат → регистар)

Директно – Direct/Absolute:

Операнд је у меморисјкој локацији.

Начини адресирања

Регистарско индиректно – Register Indirect

У регистру се налази адреса меморијске локације операнда.

Са аутоинкрементирањем – Autoincrement

Варијанта индиректног адресирања са инкрементирањем вредности регистра.

Са аутодекрементирањем – Autodecrement

Варијанта индиректног адресирања са декрементирањем вредности регистра.

Начини адресирања

Displacement/based

Задаје се константа и регистар

константа + регистар → меморијска локација операнда

Индексирано – Indexed

Задају се 2 регистра

$R0 + R1$ → меморијска локација операнда

Меморијско индиректно – Memory indirect

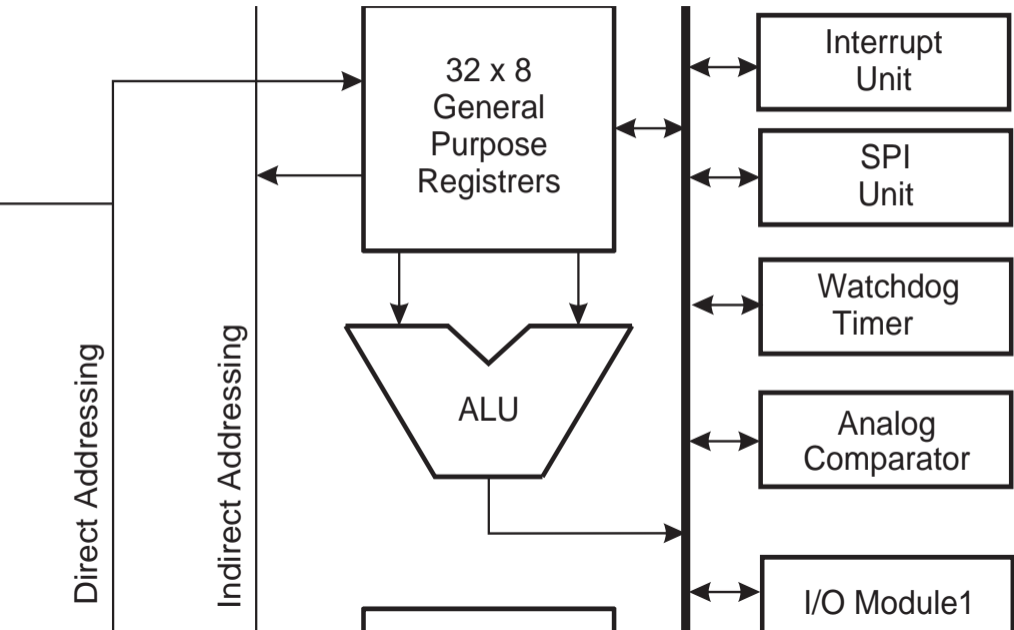
Задаје се регистар у коме се налази адреса меморијске локације која садржи адресу операнда

регистар → мем. лок. са адресом → мем. лок. операнда

Пример

Поређење адресних модова

addressing mode	example	result
immediate	ADD R1, #5	$R1 \leftarrow R1 + 5$
register	ADD R1, R2	$R1 \leftarrow R1 + R2$
direct	ADD R1, 100	$R1 \leftarrow R1 + M[100]$
register indirect	ADD R1, (R2)	$R1 \leftarrow R1 + M[R2]$
post-increment	ADD R1, (R2)+	$R1 \leftarrow R1 + M[R2]$ $R2 \leftarrow R2 + d$
pre-decrement	ADD R1, -(R2)	$R2 \leftarrow R2 - d$ $R1 \leftarrow R1 + M[R2]$
displacement	ADD R1, 100(R2)	$R1 \leftarrow R1 + M[100 + R2]$
indexed	ADD R1, (R2+R3)	$R1 \leftarrow R1 + M[R2+R3]$
memory indirect	ADD R1, @(R2)	$R1 \leftarrow R1 + M[M[R2]]$



AVR ALU

ALU је директно повезана са 32 регистра опште намене.

Инструкције се извршавају у једном циклусу такта (када су операнди у регистрима).

Инструкције:

- аритметичке
- логичке
- над једним битом

Неке имплементације подржавају и множење позитивних и негативних бројева.

AVR Регистри опште намене

7	0	Addr.
		R0 \$00
		R1 \$01
		R2 \$02
		...
		R13 \$0D
		R14 \$0E
		R15 \$0F
		R16 \$10
		R17 \$11
		...
		R26 \$1A
		R27 \$1B
		R28 \$1C
		R29 \$1D
		R30 \$1E
		R31 \$1F

Подржане комбинације:

- један 8-битни операнд и један 8-битни резултат
- два 8-битна операнда и један 8-битни резултат
- два 8-битна операнда и један 16-битни резултат
- један 16-битни операнд и један 16-битни резултат

AVR Status Register – SREG

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

I Global Interrupt Enable

V Two's Complement Overflow Flag

T Bit Copy Storage

N Negative Flag

H Half Carry Flag

Z Zero Flag

S Sign Bit, $S = N \oplus V$

C Carry Flag

AVR Показивач стека – Stack Pointer

Bit	15	14	13	12	11	10	9	8	
	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SPH
	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

- SP показује на врх стека
- мора се сетовати да показуја на адресу већу од 0x60
- инструкције: PUSH, POP (inc 1), RET (inc 2) и RETI (inc 2)

AVR Начини адресирања – Addressing Modes

Подржава више адресних модова за приступ:

- Програмској меморији – Program memory (Flash) и
 - Подацима – Data memory (SRAM, Register file, I/O Mem.).
-
- immediate/literal
 - Регистарско – Register
 - Директно – Direct/Absolute
 - Регистарско индиректно – Register Indirect
 - Са аутоинкрементирањем – Autoincrement
 - Са аутодекрементирањем – Autodecrement
 - Displacement/based
 - Индексирано – Indexed