

به نام خدا

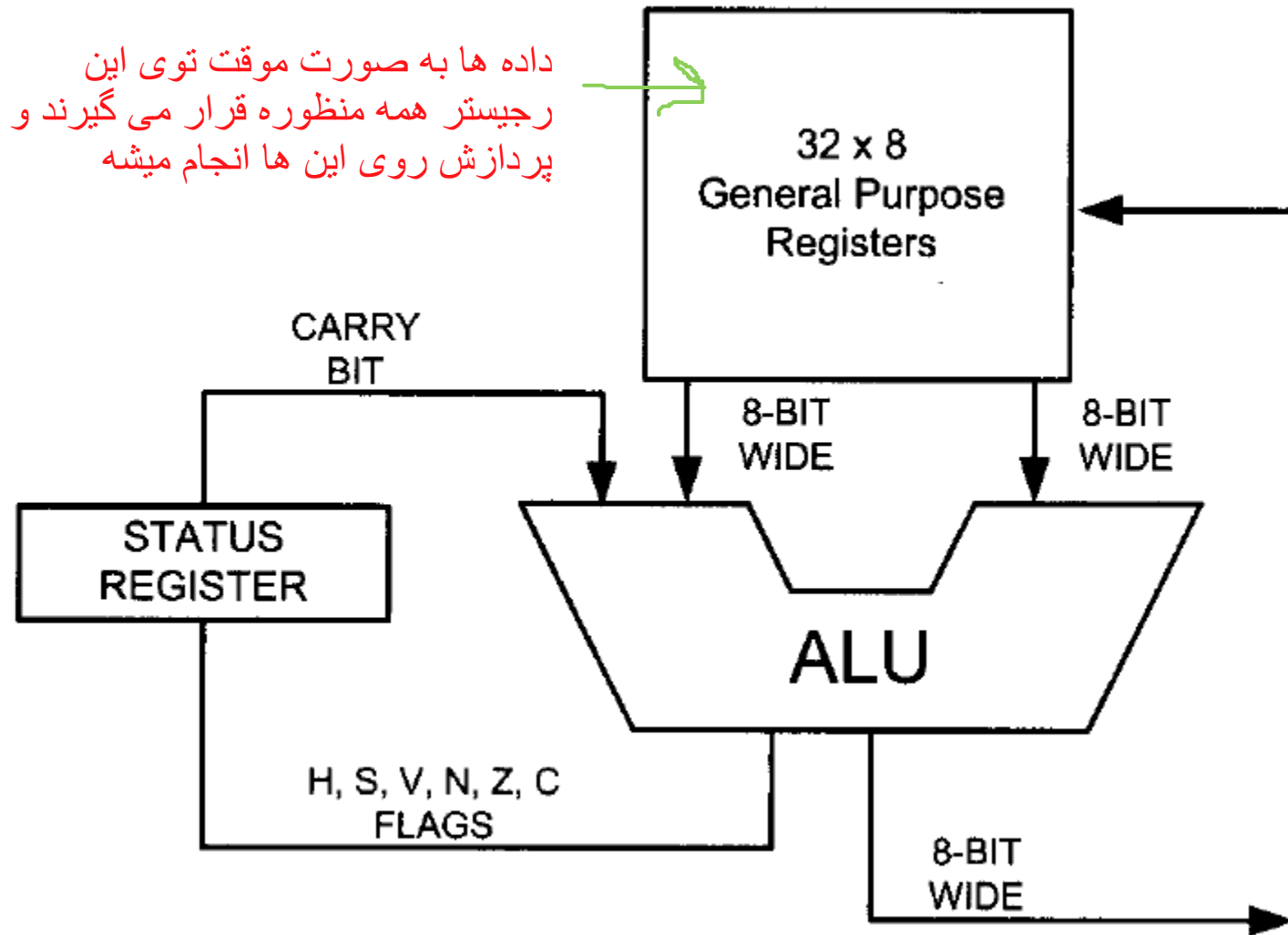
آشنایی با زبان اسمبلی AVR

دستورات پایه

Dr. Aref Karimafshar
A.karimafshar@ec.iut.ac.ir



AVR GPR and ALU



LDI – Load Immediate

این دستور 8 بیت داده رو به یک رجیستر وارد می کنه

- Loads an 8-bit constant directly to register 16 to 31

(i) $Rd \leftarrow K$

Syntax:

Operands:

Program Counter:

(i) LDI Rd,K

$16 \leq d \leq 31, 0 \leq K \leq 255$

$PC \leftarrow PC + 1$

16-bit Opcode:

1110	KKKK	dddd	KKKK
------	------	------	------

Status Register (SREG) and Boolean Formula

I	T	H	S	V	N	Z	C
–	–	–	–	–	–	–	–

Words

1 (2 bytes)

Cycles

1

MOV – Copy Register

This instruction makes a copy of one register into another. The source register Rr is left unchanged, while the destination register Rd is loaded with a copy of Rr.

Operation:

(i) $Rd \leftarrow Rr$

Syntax:

(i) MOV Rd,Rr

Operands:

$0 \leq d \leq 31, 0 \leq r \leq 31$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0010	11rd	dddd	rrrr
------	------	------	------

Status Register (SREG) and Boolean Formula

I	T	H	S	V	N	Z	C
–	–	–	–	–	–	–	–

Words 1 (2 bytes)

Cycles 1

ADD – Add without Carry

Adds two registers without the C Flag and places the result in the destination register Rd

Operation:

(i) $Rd \leftarrow Rd + Rr$

Syntax:

(i) ADD Rd,Rr

Operands:

$0 \leq d \leq 31, 0 \leq r \leq 31$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0000	11rd	dddd	rrrr
------	------	------	------

Status Register (SREG) and Boolean Formula

I	T	H	S	V	N	Z	C
–	–	\Leftrightarrow	\Leftrightarrow	\Leftrightarrow	\Leftrightarrow	\Leftrightarrow	\Leftrightarrow

Words

1 (2 bytes)

Cycles

1

ADD – Add without Carry

Status Register (SREG) and Boolean Formula

I	T	H	S	V	N	Z	C
–	–	\Leftrightarrow	\Leftrightarrow	\Leftrightarrow	\Leftrightarrow	\Leftrightarrow	\Leftrightarrow

H $Rd3 \cdot Rr3 + Rr3 \cdot \overline{R3} + \overline{R3} \cdot Rd3$

Set if there was a carry from bit 3; cleared otherwise.

S $N \oplus V$, for signed tests.

V $Rd7 \cdot Rr7 \cdot \overline{R7} + \overline{Rd7} \cdot \overline{Rr7} \cdot R7$

Set if two's complement overflow resulted from the operation; cleared otherwise.

N $R7$

Set if MSB of the result is set; cleared otherwise.

Z $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$

Set if the result is \$00; cleared otherwise.

C $Rd7 \cdot Rr7 + Rr7 \cdot \overline{R7} + \overline{R7} \cdot Rd7$

Set if there was carry from the MSB of the result; cleared otherwise.

Comment

- Use “ ; ” to write a comment in Assembly language
 - Like “//” in C language
- To present number
 - In hex
 - \$ or 0x
 - Decimal
 - No thing
 - Binary
 - 0b

LDI R16,\$50 **LDI R16,0x50**

LDI R16, 50

LDI R16,0b00110010

توی زبان اسمبلی:

برای کامنت گذاشتن از ; استفاده میشه مثل // که توی زبان سی داشتیم
اعداد رو در زبان اسمبلی اگر بخوایم در مبنای 16 یا به صورت هگز بیان کنیم می تونیم از \$ قبل
از اون بذاریم یا با 0X شروعشون کنیم مثلا میخوایم 50 هگز رو بریزیم توی R16 که میگیریم:
\$50 یا 0X50

برای اعداد در مبنای 10 و به صورت دسیمال به هیچ علامتی نیاز نداریم مثالشو ببین
و اگر بخوایم به صورت باینری بگیریم قبل از عدد 0b می داریم

NOP – No Operation

- This instruction performs a single cycle No Operation

Operation:

(i) No

Syntax:

(i) NOP

Operands:

None

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0000	0000	0000	0000
------	------	------	------

Status Register (SREG) and Boolean Formula

I	T	H	S	V	N	Z	C
–	–	–	–	–	–	–	–

Words 1 (2 bytes)

Cycles 1

این دستور کاری برای ما انجام نمی ده ینی هیچ عملیاتی انجام نمیده فقط کاری که می کنه اینه که یک سیکل ساعت می گیره

در شکل سمت چپ قبل از اینکه دستور ADD را بزنین سیکل ساعت 3 هست ولی زمانی که روی دستور ADD می ریم سیکل ساعت 16384 میشه بخاطر این است که بعد از انجام دستور ADD این برنامه مشخص نیست که داره تموم میشه پس سراغ همه خونه های حافظه بعدی هم می ره و همینجور ادامه پیدا میکنه و می ره جلو تا دیگه اپکدی نباشه پس برای اینکه جلوی این کار رو بگیریم از دستور NOP استفاده میکنیم (شکل سمت راست)

```
LDI R16,0x55
LDI R17,0b00000001
MOV R18,R17
ADD R17,R16
NOP
```

100 %

Name	Value
Program Counter	0x00000000
Stack Pointer	0x0000
X Register	0x0000
Y Register	0x0000
Z Register	0x0000
Status Register	11111111
Cycle Counter	16384

```
LDI R16,0x55
LDI R17,0b00000001
MOV R18,R17
ADD R17,R16
NOP
```

100 %

Name	Value
Program Counter	0x00000004
Stack Pointer	0x0000
X Register	0x0000
Y Register	0x0000
Z Register	0x0000
Status Register	11111111
Cycle Counter	4

ADC – Add with Carry

- Adds two registers and the contents of the C Flag and places the result in the destination register Rd

Operation:

(i) $Rd \leftarrow Rd + Rr + C$

Syntax:

(i) ADC Rd,Rr

Operands:

$0 \leq d \leq 31, 0 \leq r \leq 31$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0001	11rd	dddd	rrrr
------	------	------	------

Status Register (SREG) and Boolean Formula

I	T	H	S	V	N	Z	C
–	–	\Leftrightarrow	\Leftrightarrow	\Leftrightarrow	\Leftrightarrow	\Leftrightarrow	\Leftrightarrow

Words 1 (2 bytes)

Cycles 1

دستور جمع با استفاده از کری:

توی این نوع از جمع علاوه بر اینکه مقدار دو رجیستر با هم جمع میشه کری هم حساب میشه ینی اگر کری یک باشه به مجموع این ها اضافه خواهد شد

شیوه این تاثیر روی پرچم ها مثل همون چیزی است که درباره دستور add گفتیم ینی همون فرمولا رو داره

SUB – Subtract Without Carry

- Subtracts two registers and places the result in the destination register Rd

Operation:

(i) $Rd \leftarrow Rd - Rr$

Syntax:

(i) SUB Rd,Rr

Operands:

$0 \leq d \leq 31, 0 \leq r \leq 31$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0001	10rd	dddd	rrrr
------	------	------	------

Status Register (SREG) and Boolean Formula

I	T	H	S	V	N	Z	C
–	–	\Leftrightarrow	\Leftrightarrow	\Leftrightarrow	\Leftrightarrow	\Leftrightarrow	\Leftrightarrow

Words 1 (2 bytes)

Cycles 1

دستور تفریق بدون کری:

SUB – Subtract Without Carry

H $\overline{Rd3} \cdot Rr3 + Rr3 \cdot R3 + R3 \cdot \overline{Rd3}$

Set if there was a borrow from bit 3; cleared otherwise.

S $N \oplus V$, for signed tests.

V $Rd7 \cdot \overline{Rr7} \cdot \overline{R7} + \overline{Rd7} \cdot Rr7 \cdot R7$

Set if two's complement overflow resulted from the operation; cleared otherwise.

N $R7$

Set if MSB of the result is set; cleared otherwise.

Z $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$

Set if the result is \$00; cleared otherwise.

C $\overline{Rd7} \cdot Rr7 + Rr7 \cdot R7 + R7 \cdot \overline{Rd7}$

Set if the absolute value of the contents of Rr is larger than the absolute value of Rd; cleared otherwise.

فرمول رجیسترهای وضعیت برای تفریق به صورت روبرو است:

SBC – Subtract with Carry

- Subtracts two registers and subtracts with the C Flag, and places the result in the destination register Rd.

Operation:

(i) $Rd \leftarrow Rd - Rr - C$

Syntax:

(i) SBC Rd,Rr

Operands:

$0 \leq d \leq 31, 0 \leq r \leq 31$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0000	10rd	dddd	rrrr
------	------	------	------

Status Register (SREG) and Boolean Formula

I	T	H	S	V	N	Z	C
–	–	\Leftrightarrow	\Leftrightarrow	\Leftrightarrow	\Leftrightarrow	\Leftrightarrow	\Leftrightarrow

Words 1 (2 bytes)

Cycles 1

دستور تفریق با کری:
اگر کری فعال بود از این دستور استفاده میشه

AND – Logical AND

- Performs the logical AND between the contents of register Rd and register Rr, and places the result in the destination register Rd

Operation:

(i) $Rd \leftarrow Rd \cdot Rr$

Syntax:

(i) AND Rd,Rr

Operands:

$0 \leq d \leq 31, 0 \leq r \leq 31$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0010	00rd	dddd	rrrr
------	------	------	------

Status Register (SREG) and Boolean Formula

I	T	H	S	V	N	Z	C
–	–	–	\Leftrightarrow	0	\Leftrightarrow	\Leftrightarrow	–

Words 1 (2 bytes)

Cycles 1

دستورات منطقی:

دستور AND:

مقدار دوتا رجیستر رو که میتونه یکی از اون رجیسترهای همه منظوره باشه با هم به صورت منطقی و بیت به بیت AND میکنه
S, N , Z پرچم هایی هستند که با اجرای این دستور تحت تاثیر قرار میگیرند

AND – Logical AND

S $N \oplus V$, for signed tests.

V 0

Cleared.

N R7

Set if MSB of the result is set; cleared otherwise.

Z $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$

Set if the result is \$00; cleared otherwise.

فرمول های رجیستر وضعیت:

OR – Logical OR

- Performs the logical OR between the contents of register Rd and register Rr, and places the result in the destination register Rd

Operation:

(i) $Rd \leftarrow Rd \vee Rr$

Syntax:

(i) OR Rd,Rr

Operands:

$0 \leq d \leq 31, 0 \leq r \leq 31$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0010	10rd	dddd	rrrr
------	------	------	------

Status Register (SREG) and Boolean Formula

I	T	H	S	V	N	Z	C
–	–	–	\Leftrightarrow	0	\Leftrightarrow	\Leftrightarrow	–

Words 1 (2 bytes)

Cycles 1

دستور OR: بیت به بیت با هم دیگه OR میکنه

EOR – Exclusive OR

- Performs the logical EOR between the contents of register Rd and register Rr and places the result in the destination register Rd

Operation:

(i) $Rd \leftarrow Rd \oplus Rr$

Syntax:

(i) EOR Rd,Rr

Operands:

$$0 \leq d \leq 31, 0 \leq r \leq 3$$

Program Counter:

$$PC \leftarrow PC + 1$$

16-bit Opcode:

0010	01rd	dddd	rrrr
------	------	------	------

Status Register (SREG) and Boolean Formula

I	T	H	S	V	N	Z	C
–	–	–	\Leftrightarrow	0	\Leftrightarrow	\Leftrightarrow	–

Words 1 (2 bytes)

Cycles 1

دستور EOR: دو تار جیستر رو با هم دیگه xor میکنه

Summary

ALU Instructions Using Two GPRs

Instruction		
ADD	Rd, Rr	ADD Rd and Rr
ADC	Rd, Rr	ADD Rd and Rr with Carry
AND	Rd, Rr	AND Rd with Rr
EOR	Rd, Rr	Exclusive OR Rd with Rr
OR	Rd, Rr	OR Rd with Rr
SBC	Rd, Rr	Subtract Rr from Rd with carry
SUB	Rd, Rr	Subtract Rr from Rd without carry

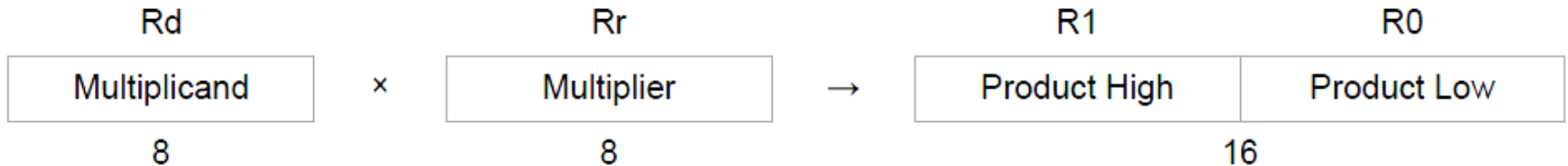
ADD	Rd, Rr	ADD Rd and Rr
ADC	Rd, Rr	ADD Rd and Rr with Carry
AND	Rd, Rr	AND Rd with Rr
EOR	Rd, Rr	Exclusive OR Rd with Rr
OR	Rd, Rr	OR Rd with Rr
SBC	Rd, Rr	Subtract Rr from Rd with carry
SUB	Rd, Rr	Subtract Rr from Rd without carry

Rd and Rr can be any of the GPRs.

این دستورات دو عملوندی هستند

MUL – Multiply Unsigned

- This instruction performs 8-bit \times 8-bit \rightarrow 16-bit unsigned multiplication



- The multiplicand *Rd* and the multiplier *Rr* are two registers containing unsigned numbers. The 16-bit unsigned product is placed in *R1* (high byte) and *R0* (low byte). Note that if the multiplicand or the multiplier is selected from *R0* or *R1* the result will overwrite those after multiplication.

بعضی از نسخه های میکروکنترلر خانواده avr از دستور ضرب هم پشتیبانی می کنند
دستور ضرب: یک دستوری هستش که دو تا عملوند 8 بیتی به عنوان ورودی داره و خروجی اون
یک خروجی 16 بیتی هستش

از دستور MUL استفاده میکنیم که به صورت اعداد بی علامت میاد ضرب رو انجام میده
عملوند اول ینی Rd و عملوند دوم ینی Rr در هم ضرب میشن و جواب در رجیستر های R1, R0
قرار میگیره و قسمت کم ارزش جواب در رجیستر R0 قرار میگیره و قسمت پر ارزشش در
رجیستر R1 قرار می گیره

این Rr , Rd می تونن یکی از اون رجیسترهای همه منظوره 32 باشند
و اگر این دوتا ورودی هم یکی از این R0 یا R1 باشه جواب داخلش overwrite میشه

متن اسلاید:

ضریب Rd و ضریب Rr دو رجیستر حاوی اعداد بدون علامت هستند. محصول بدون علامت 16
بیتی در R1 (بایت بالا) و R0 (بایت کم) قرار می گیرد. توجه داشته باشید که اگر ضرب یا
ضریب از R0 یا R1 انتخاب شود، نتیجه پس از ضرب، آنها را بازنویسی می کند.

MUL – Multiply Unsigned

Operation:

(i) $R1:R0 \leftarrow R_d \times R_r$ (unsigned \leftarrow unsigned \times unsigned)

Syntax:

(i) MUL R_d, R_r

Operands:

$0 \leq d \leq 31, 0 \leq r \leq 31$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	11rd	dddd	rrrr
------	------	------	------

Status Register (SREG) and Boolean Formula

I	T	H	S	V	N	Z	C
–	–	–	–	–	–	\Leftrightarrow	\Leftrightarrow

C R15

Z $\overline{R15} \cdot \overline{R14} \cdot \overline{R13} \cdot \overline{R12} \cdot \overline{R11} \cdot \overline{R10} \cdot \overline{R9} \cdot \overline{R8} \cdot \overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$

Set if the result is \$0000; cleared otherwise.

فرمت دستور:

MULS – Multiply Signed

- This instruction performs 8-bit × 8-bit → 16-bit signed multiplication

Operation:

(i) $R1:R0 \leftarrow R_d \times R_r$ (signed ← signed × signed)

Syntax:

(i) MULS R_d, R_r

Operands:

$16 \leq d \leq 31, 16 \leq r \leq 31$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0000	0010	dddd	rrrr
------	------	------	------

Status Register (SREG) and Boolean Formula

I	T	H	S	V	N	Z	C
–	–	–	–	–	–	⇔	⇔

C R15

Z $\overline{R15} \cdot \overline{R14} \cdot \overline{R13} \cdot \overline{R12} \cdot \overline{R11} \cdot \overline{R10} \cdot \overline{R9} \cdot \overline{R8} \cdot \overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$

Set if the result is \$0000; cleared otherwise.

دستور ضرب با علامت:

پایان

موفق و پیروز باشید