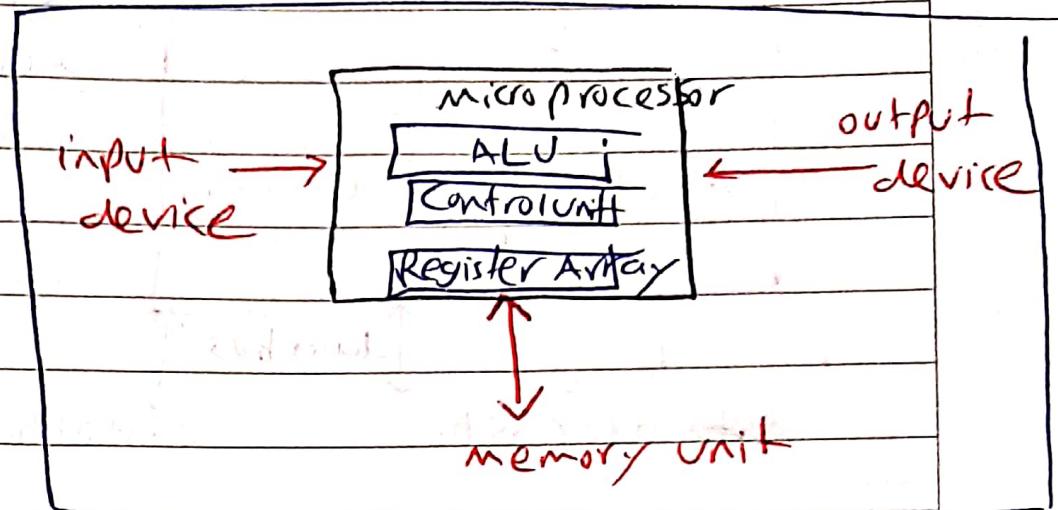


my hand writings
during microprocessor course in INST university
hadi ghafori

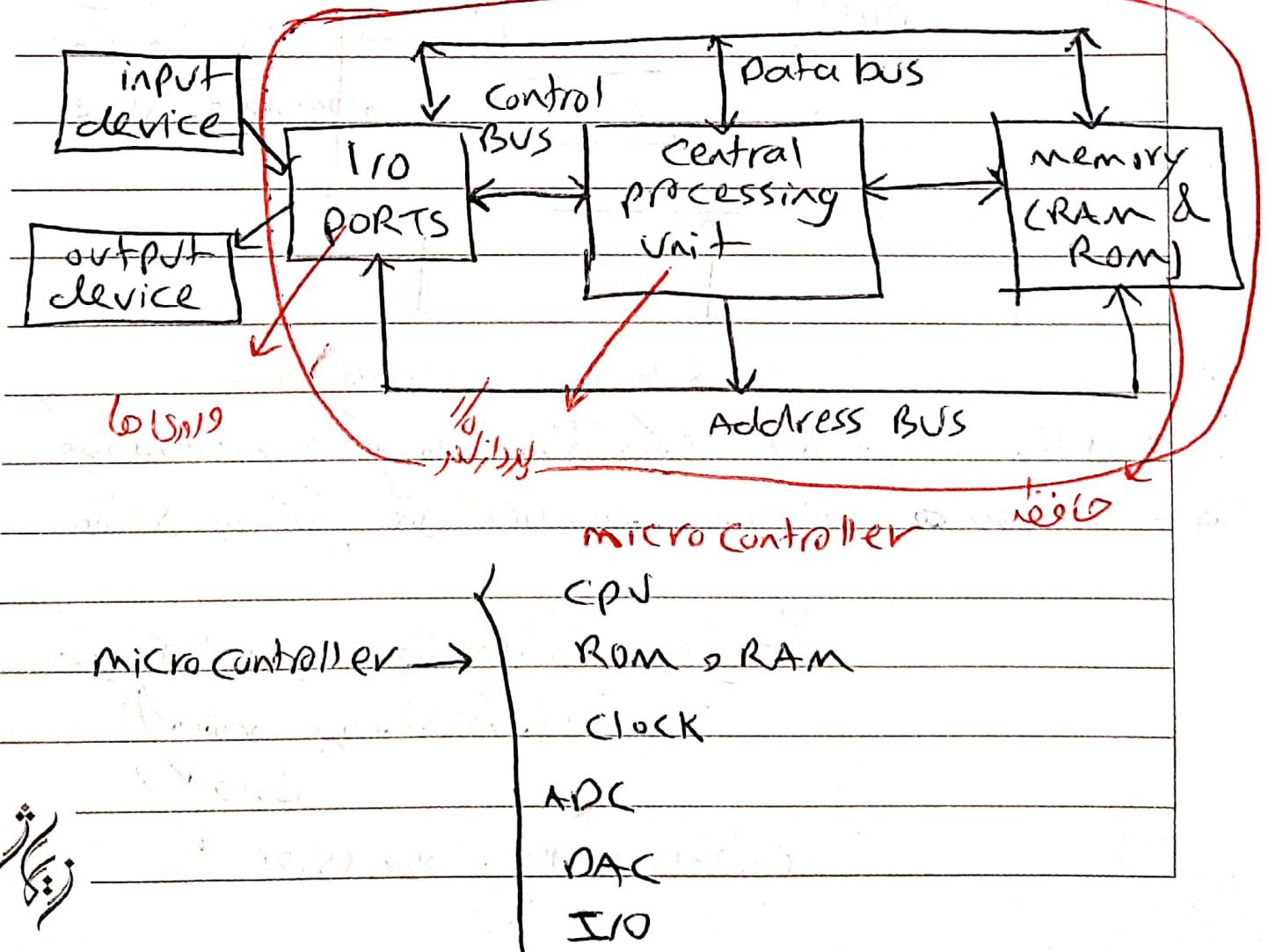
◀ DRAFT ▶

session 7)

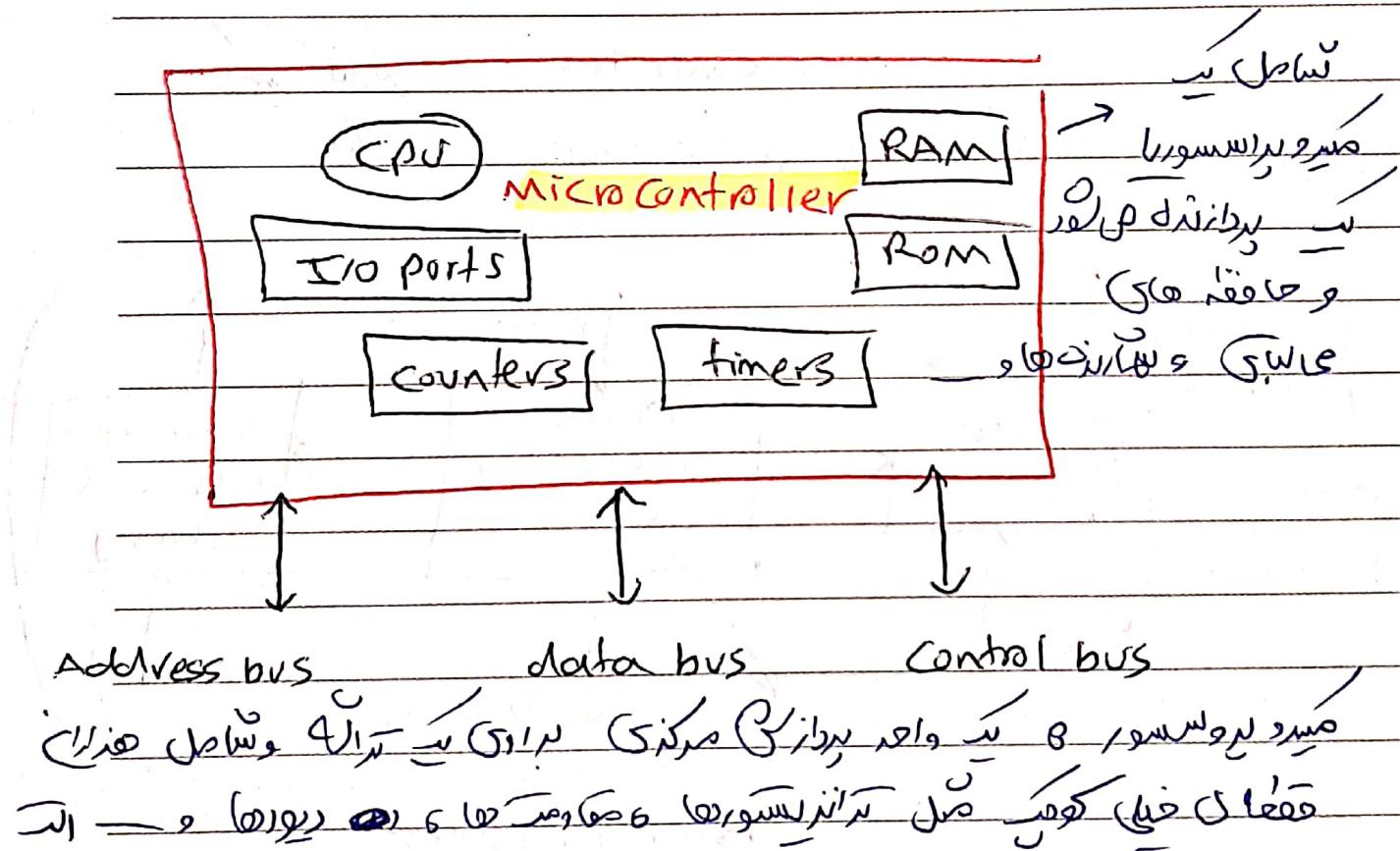
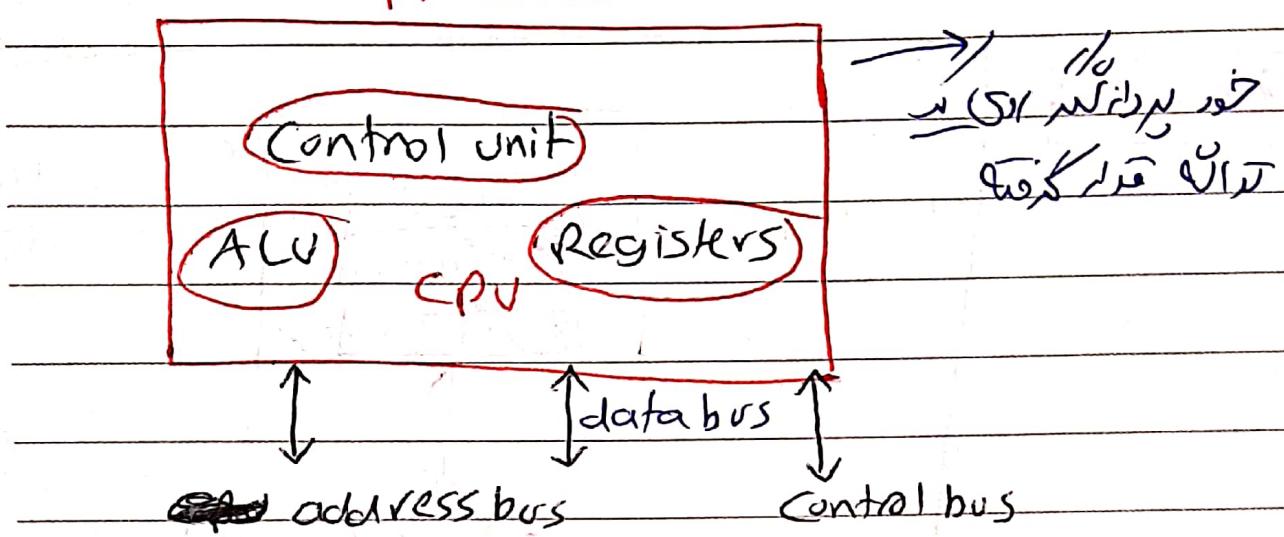


micro controller

الجهاز المركب (Microcontroller) هو جهاز مركب يحتوي على معالج (CPU)، ذاكرة (Memory)، واجهات إدخال/إخراج (I/O Ports).



Microprocessor



مقدار دفعات (cycles) كثيف (fast) و متساهم (slow) معاً (fast and slow together)

مقدار دفعات (cycles) بطيء (slow) معاً (slow and slow together)

الآن (now) المعاشر (nearby) والجوار (neighbor)



Micro controller vs micro processor

Micro processor	Micro controller
العالي	المنخفض
Advanced data processing	Simple data processing
video, computer vision,	audio, sensor processing
personal computers, fast	embedded systems
communications, multi-core	Embedded devices
computation	control systems, smartphones
higher power consumption	lower power consumption
higher cost	lower cost
higher performance	lower performance
higher clock speed	lower clock speed
more memory	less memory
more peripherals	fewer peripherals
I/O pins	programmable digital and analog I/O pins

65 MP (ادی)

(65 MP ادی) intel 4004 * ①

1971 *

4-bit central processing unit *

1μs (پلیس نوی دلیل) *

480 KHz 8 μs *

40250 to 92500 instruction per second

②

Intel 8008

1972

8-bit central processing unit

first commercial non calculating personal computers

5μs to 200 KHz to 800 KHz

35000 to 80000 instruction per second

③

IMP-16

① 1973

② 16-bit central processing unit

③ microcomputers

④ 715 KHz

1979

④ Motorola 68000 →

32 bit central processing unit
new generation of personal
computers

8-20 MHz

⑤ R4000 →

1991

② 64 bit central processing unit
③ new generation of personal
computers
④ 700, 733, 750, 200, 250 MHz

16 bit microprocessor 8

① 8086 4.7 MHz, 8 MHz, 10 MHz
② 8088 8 MHz
③ 80186 / 80188 8.6 MHz
④ 80286 16 MHz

32 bit microprocessor 8

① Intel 80386 8-16 MHz to
33 MHz
② Intel 80486 16 MHz to
100 MHz
③ pentium 8-60 MHz

64 bit micro processor

- ① Intel core 2 @ 7.2 GHz to 3 GHz
- ② Intel i7 @ 2.05 GHz to 3.33 GHz
- ③ ~ i5 @ 2.4 GHz to 3.6 GHz
- ④ ~ i3 @ 2.93 GHz to 3.33 GHz

مقدار صیغه دستورالعمل و کد نزدیکی

چون هنوز هم از قابلیت ۴۸ بت فواید می‌باشد اینجا در حد ۳ سال ۰۲

به همای باز نگذارید

از ۴۸ بت تا ۳۲ بت تا ۱۶ بت تا ۸ بت تا ۴ بت

لرچ صیغه دستورالعمل خواهد بود

Reduced instruction set computer (RISC)

ارفاع صیغه دستورالعمل

DEC Alpha 6 MC88100 6 IBM RS 8 چیز

4000

pentium...

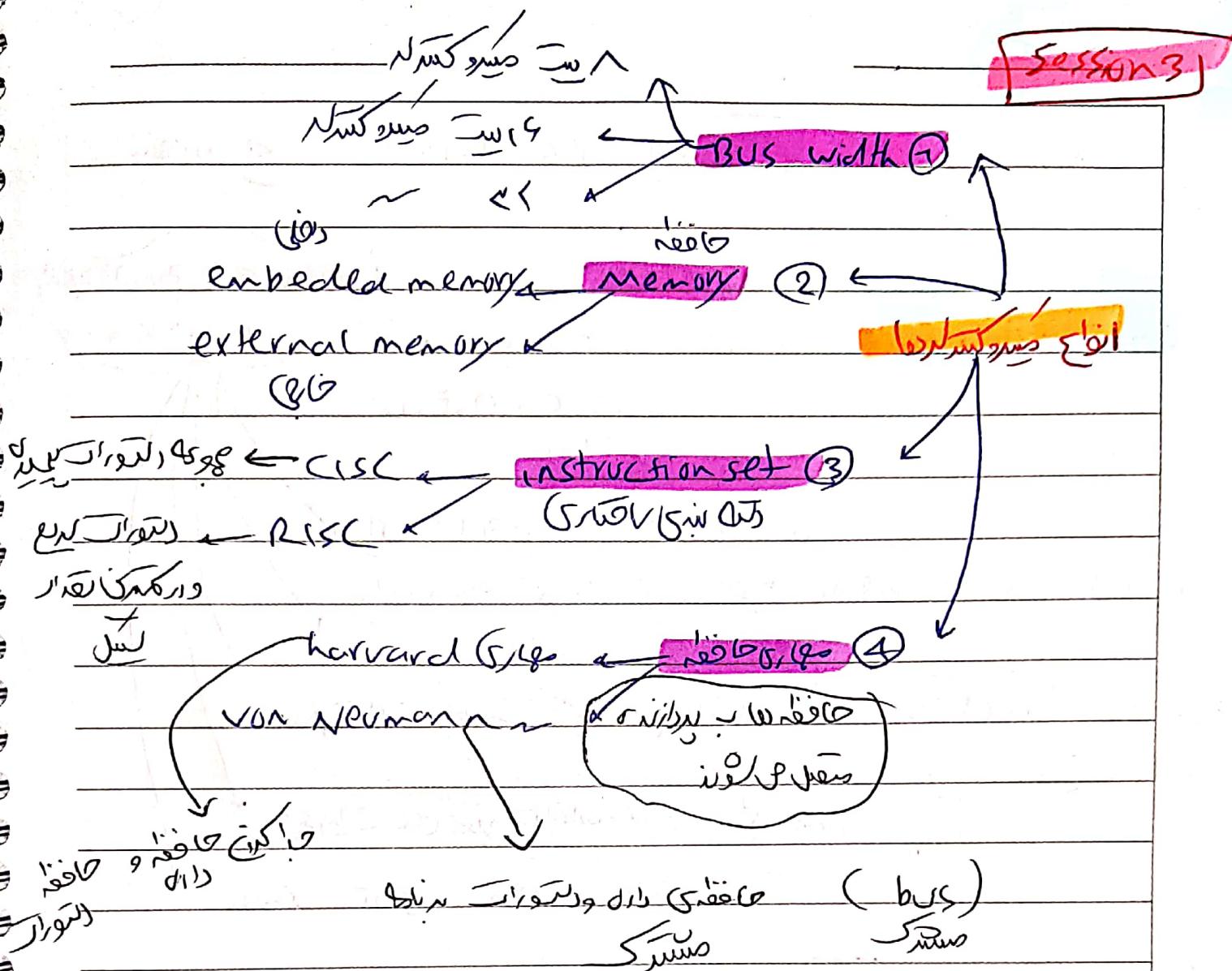
Complex instruction set computer (CISC)

اجام کردن دستورات ایجاد کردن دستورات

Intel 486

Intel 386

SESSION 3



SESSION 1

Accumulator → (累加器)

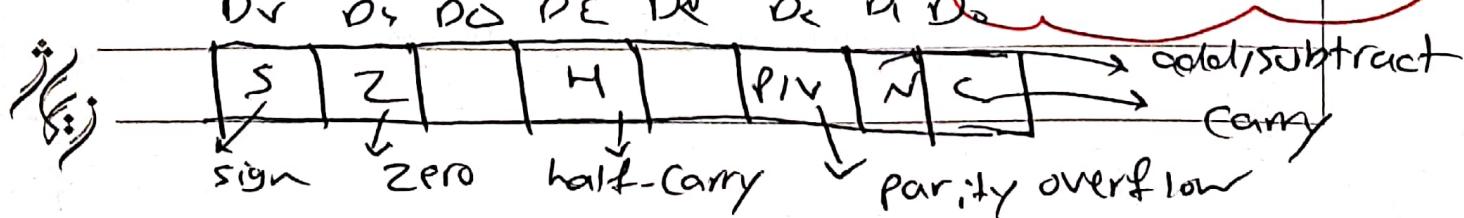
累加器和结果 (累加器和结果)

累加器 (累加器) 是一个临时寄存器，用于存储中间结果 (中间结果)。累加器 (累加器) 通常与结果 (结果) 一起使用。

累加器 (累加器) + 累加器 (累加器) = ADD → 累加器 (累加器)

累加器 (累加器) - 累加器 (累加器) = AC (累加器)

Flag registers



محلل این register هنوز ALU یا register نیست

General purpose registers

280 مکانیزم register های General purpose می باشد

B, C, D, E, H, L

این های register های General purpose هستند

برای ذخیره ازای داده های اینها صور التقدیم قدری کمتر

مکانیزم Register Transfer هم این تقاردها را درین میان رساند

از بین این مکانیزم

copy, load و store این تقاردها را درین مکانیزم میگذرانند

ALU یا 280 Port Bus

Accumulator A	Flag F
B	C
D	E
H	L

F' A'

اینها flag و AC های

Alternate registers

280 مکانیزم register دارد

B', C', D', E', H', L'

که اینها جای خواهی (جای خواهد) قرار گیرند

لیکن اینها قابل (جای خواهی) نیستند

زیکر

Index registers

280

(العنوان)

280 (العنوان) + (العنوان)

IX, IY

displacement (النسبة)

count (العداد)

مثلاً لو كان العنوان 2050H ، و IX = 2050H ، فـ displacement = 0 ، count = 10H

displacement count ~ [10H] ، با (نسبة كون صفر ، العدد كون صفر)

مثلاً لو كان العنوان 2040H ، و IX = 2040H ، فـ displacement = 0 ، count = 10H

[10H] ، با (نسبة كون صفر ، العدد كون صفر)

pointer (العنوان) ، memory pointer (العنوان) ، HL (العنوان)

memory pointer (العنوان) ، DT ، BC ، *

displacement (النسبة)

content (العنوان) ، HL (العنوان)

displacement byte

مشتر

SP

280

Stack pointer

read / write

in (P), in stack (S)

(W) in (W)

in R/W memory کر کے از طرفہ سنیں stack

stack (S) کو اس کری (از طرفہ) ← stack (S)
pointer متنقیل کرو

PC

program counter

(W) in (W)

امنیت کر کے روی تھفہ و اچانک تغیرات صوابی
اپنے بے (تغیراتی) تغیرات ایجاد کرو

کر کے توی ایسی حالت قرارداد نہ رکھی بلکہ ایسی قدر کر

واز ادکانی تو اسی دستورات مولتی پلٹر را توی Bus کی قدر بینے کر
بے سیمی صفت آئندہ کر دستورات را دیکھ کر

I

Interrupt vector register

به ما کسی مکان کرنا

ایسی حالت در فردا

in (W)

in (W)

وہی سے دستور خروجی کو سیگنال سو ایجاد

کر دستورات اخباری کا درستی را بینے کر کر

سیگنال سو ایجاد کرنا

ایسی حالت در فردا کو از طرفہ برقرار کری را زنده کرنا

و مکاری بیان تولٹ دستوری کر دستورات اخباری کو کر کر

ایجاد کرو

R J

Memory refresh

register

(W R M)

(D) pointer (W D, S, B, C, L)

(W D, S, B, C, L) (W D, S, B, C, L) refresh & write
is done

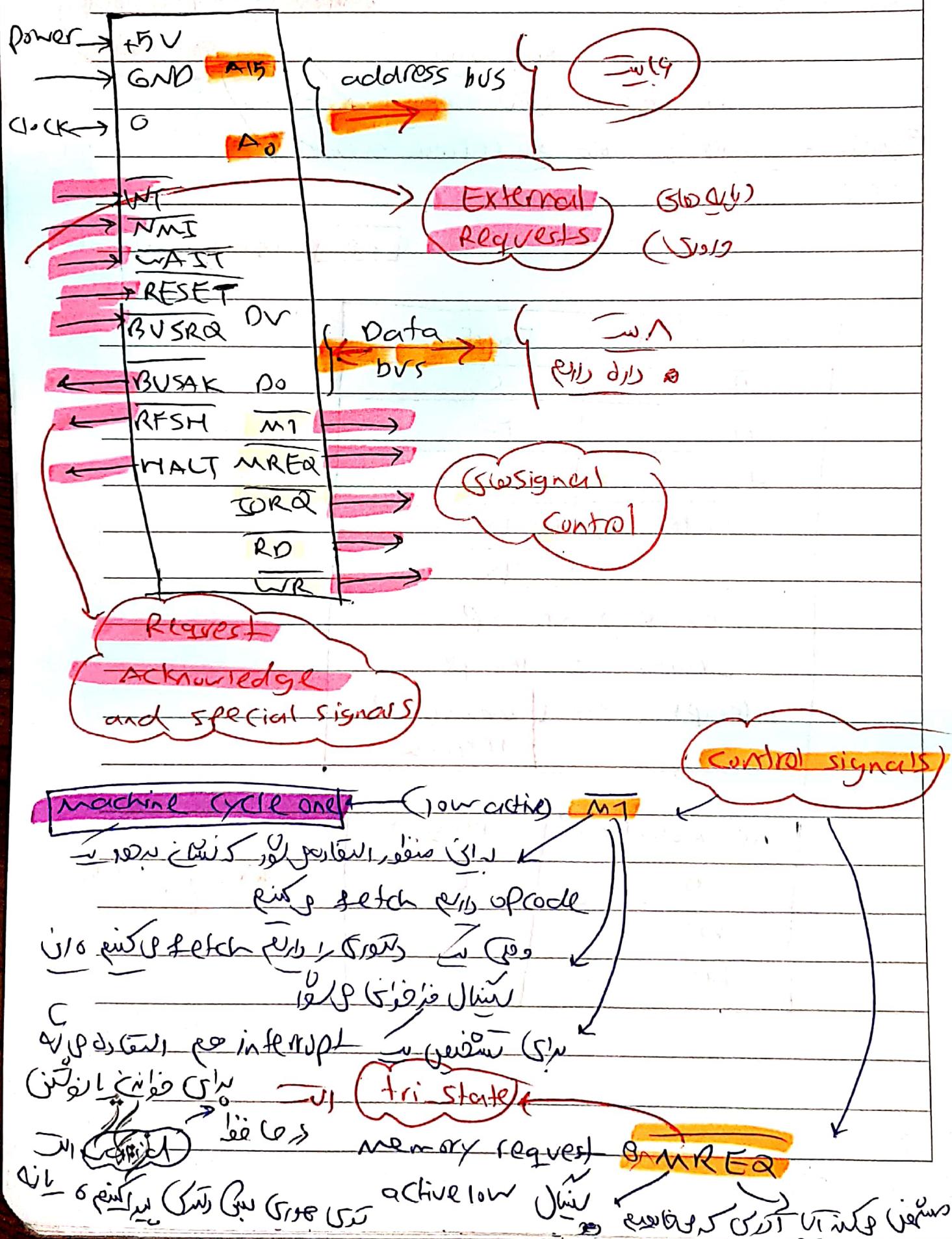
280

(W register)

Accumulator A	Flag F
B	C
D	E
H	L
Index register IX	
~ ~ IY	
Stack pointer SP	
Program Counter PC	
Interrupt vector I Memory R refresh	

Accumulator A'	Flags F'
B'	C'
D'	E'
H'	L'

Alternate flags



IORQ

active low $\bar{I} \bar{O} R Q$ I/O Request

all tri-state

العنوان (address) كـ (عنوان مركب) address bus

(A₁₇-A₀) \rightarrow الـ address bus

العنوان (address) bus

new interrupt

Read

IORQ

tri-state, active low

الـ (read) bus

MEMRD

all tri-state

I/O

MREQ

الـ (read) bus

IORQ

IORD

all tri-state

active low, tri-state

write

WR

الـ (write) bus

IOWR

memory request, write bus

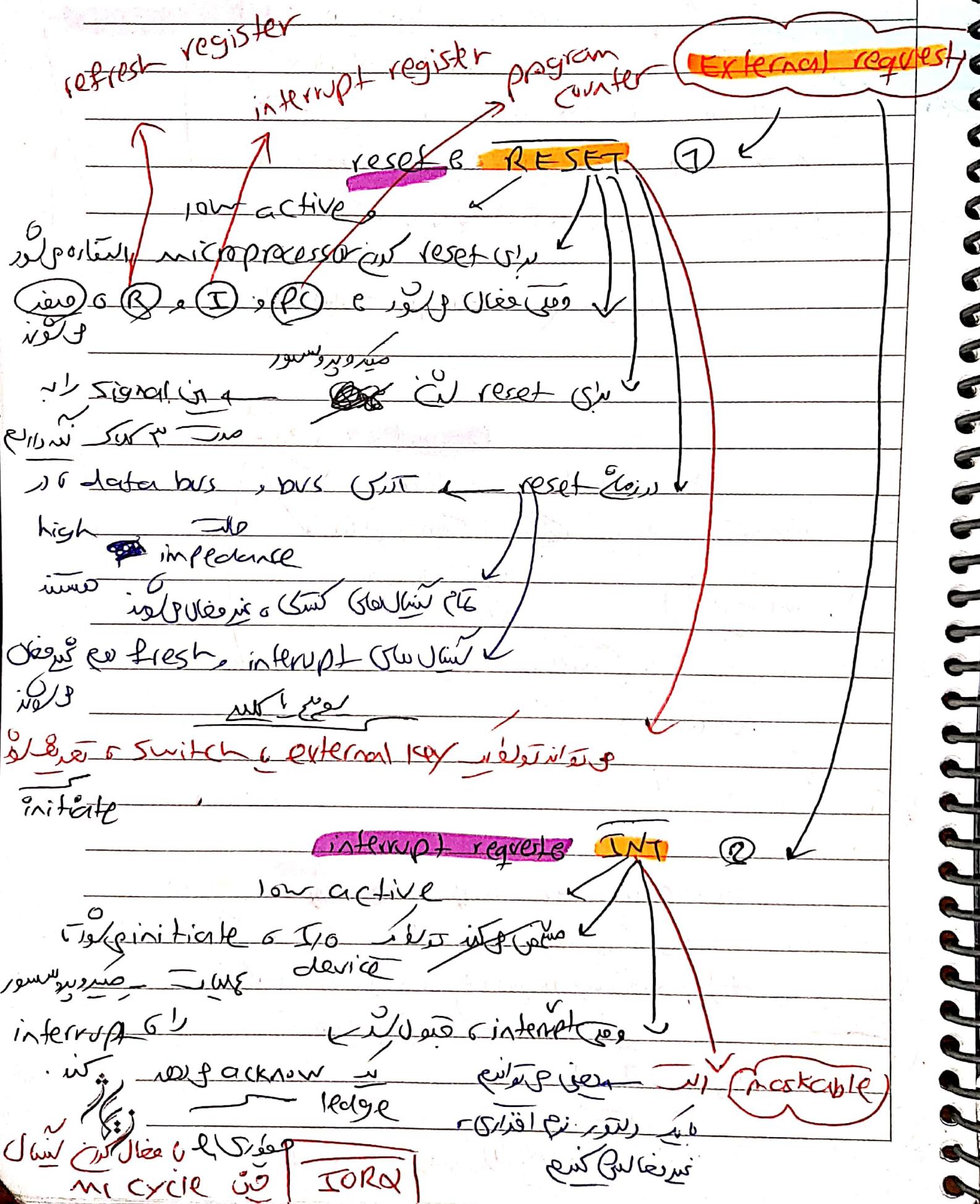
IOWR

الـ (write) bus

SIOWR

(MEMWR)

memory write



Non maskable interrupt

8 NMI ③

پر لور غیر قابل برگردان

برای وقفه های که خنی می دارند و این فرآورده ها نیز
از این لئنل را ایجاد نموده اند ای این لئنل را Pin signal معنی دارند

acknowledge

negative edge

از لور
triggered

تولید

از لور می خویم و مغایر باشیم

DMA

Direct memory access controller

bus request

BUSREQ

⊕

active low

از لور

زنای کس

I/O

لئنل را فراخوانی کن و عطف کاری کن و فتح شوند

با این برابری

اول برمگشته و در اختصار بروانده می گذرد

bus ready I/O می تواند سیگنال low را به

عنوانی تبدیل کند از آنکه BUS و می تواند

امانات bus ایجاد کند اینکه درجه زیادی

control signals

I/O

memory

برای این مسکن

wait

WAIT

⑤

active low signal

زنای کس

I/O

این فرآورده ای که داشتم

کند از خانی بگرد که هر کسی فرضی است

ب مقایله ای این سیگنال هم 280

که قدر بود این دارای

با اتفاقه کردن

نحوه چهارم

مهیا

باعظ از زمان

ملایر

**BUS acknowledge
by special
signals**

BUS RQ bus acknowledge **BUSAK** ①

low active

bus request (جهاز يطلب ارسال بوس)

جهاز يطلب

bus acknowledge (جهاز يطلب ارسال بوس)

control signals , data bus , address bus

idle state (حاله فارغه) high impedance

bus request (جهاز يطلب ارسال بوس)

RD, **WR**, **MREQ**, **IORD**

micro processor

halt & **HALT** ②

active low

halt, microprocessor

inv

refresh & **RFSH** ③

active low

(A0-A7)

bus (جهاز يطلب ارسال بوس)

refresh (جهاز يطلب ارسال بوس)

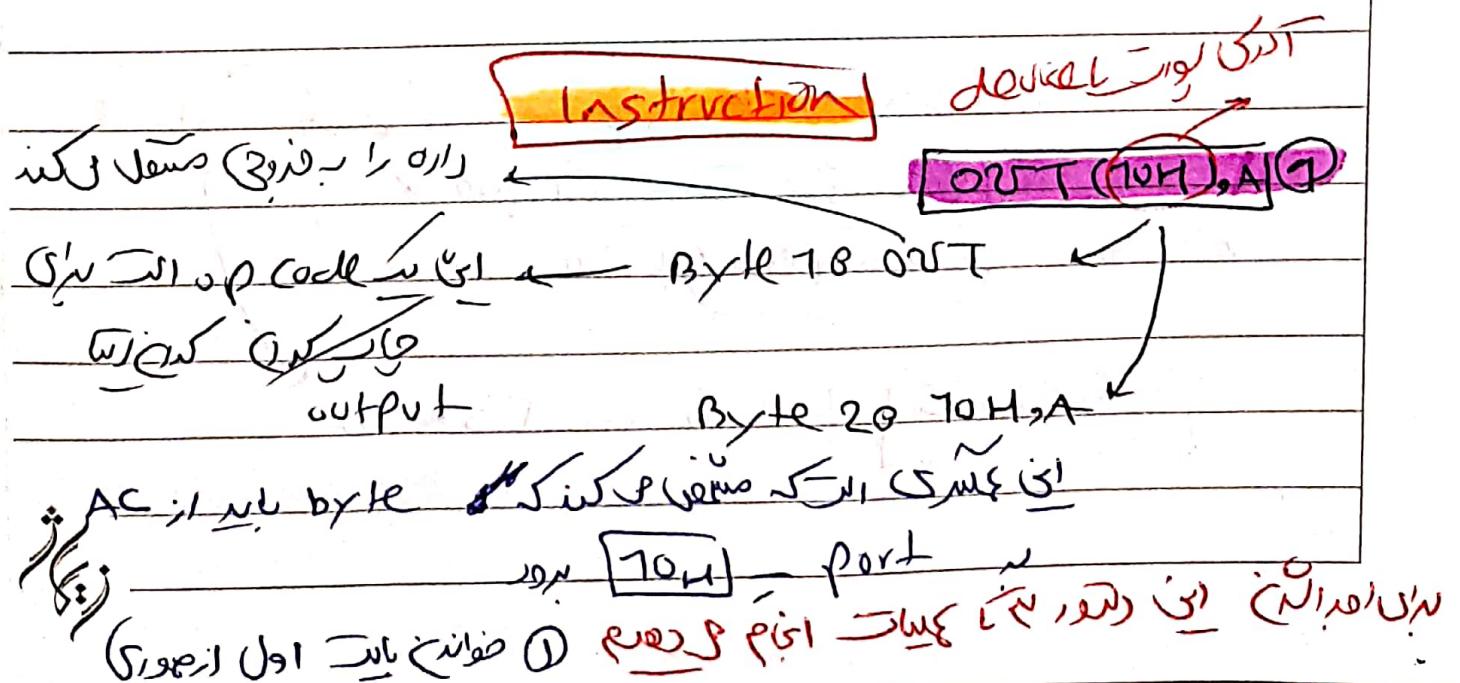
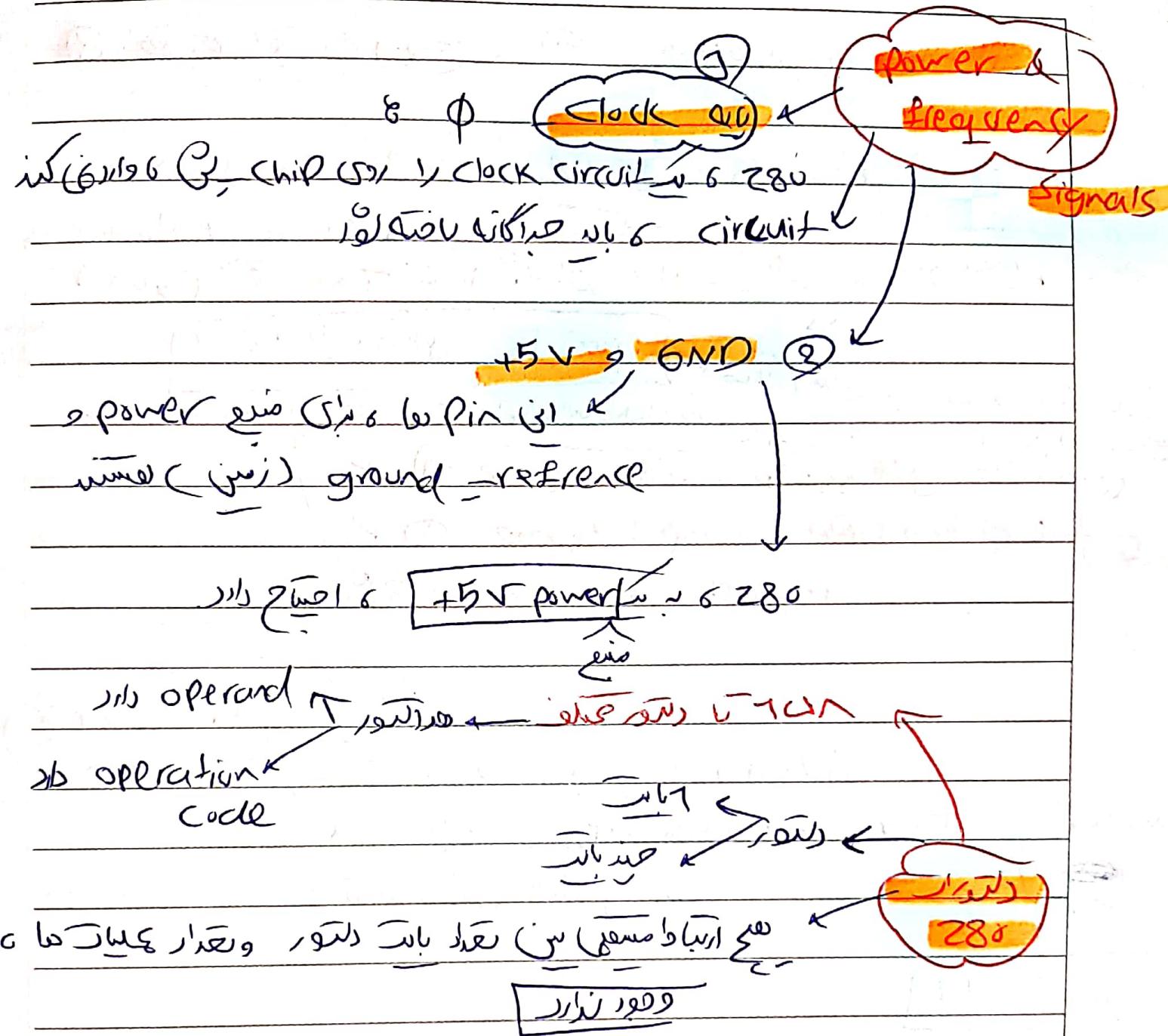


MREQ

refresh (جهاز يطلب ارسال بوس)

bus (جهاز يطلب ارسال بوس)

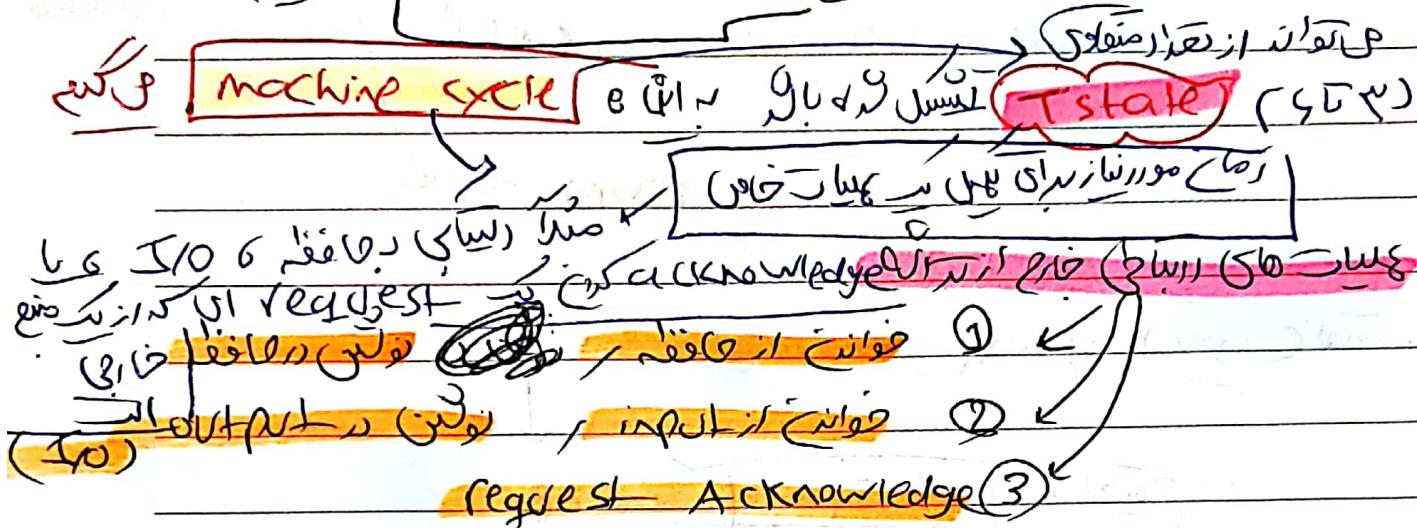
refresh (جهاز يطلب ارسال بوس)



(10H)

۳) حواسی بایت رو ارجوی و ۴) ارسال داده به پورت صورت نظر

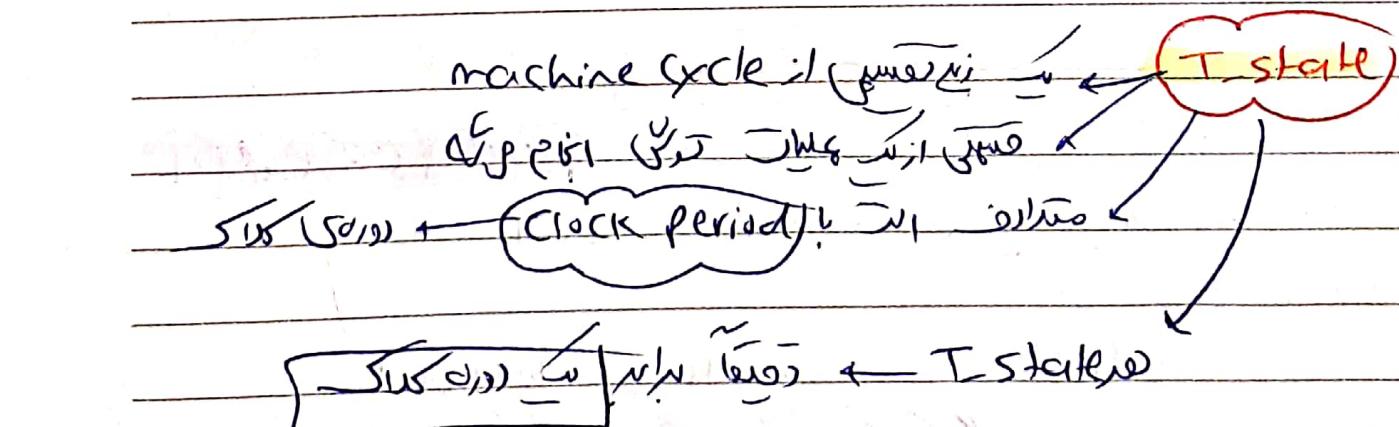
هر دستور ۲۸۰ میلی ثانیه (ms) زیرا basic (۵۰۰ ms) است



Instruction cycle

machine cycle

instruction



زیارت

Machine Cycle	M1	MREQ	IDRQ	RD	CWR
opcode fetch (M1)	0	0	1	0	1
memory read	1	0	1	0	1
~ write	1	0	1	1	0
I/O Read	1	1	0	0	1
~ write	1	1	0	1	0
interrupt Acknowledge	0	1	0	1	1
non-maskable interrupt	0	1	0	1	1
BUS ACKNOWLEDGE	0	0	1	0	1
(BUSAK) = 0	1	2	2	2	2

active ← into

inactive ← out

high impedance →

opcode fetch
machine cycle

اعتبر المدخلات بـ 280 اعترف بالخط بـ 280
(جهاز) راجعه (جهاز) اجهزة (جهاز) اجهزة

العنوان يدخل من هنا

(program counter) PC

عنوان يدخل من هنا

تاریخ رایانی کن

decoding

circuit خارجی

تولید کننده دستورات PC (جی کرک) می باشد (۳)
و در هر قسم از دستورات ۲۰۲۱ می باشد

(G RD و MREQ (G0 Signal ۲۸۰) (۴)

دستورات G0، bus (جی اند بوس) از دستورات معمولی می باشد

با فرمان (G0)

memery output

input

(opcode ۴۷H)

جی اند مموری register (جی کرک) می باشد (۵)

جی اند data bus (جی دی بوس)

و معمولی است که فرمانات را می پسند

دستورات معمولی

جی اند و دستور را اجرا کن و دستور G ۲۸۰ (opcode ۶ ۲۸۰) (۶)

input (B)، register (جی کرک) AC (جی ای سی) (۷)

AL (جی او ال) → operand

LD (جی ال دی) B (جی بی) A (opcode) (جی او کوڈ)
AC (جی ای سی)

مراحل خوبین

memory read machine cycle

Job e address machine code instruction

2000H 00111110 → ۳E LDA A, 9FH

2001H 70011111 → ۹F

، ۹FH را می بینیم (۸)

زیرگذشت

لکھوں (۹)

لکھوں (۱۰)

، op code ، و مول 6 cycle (T_E)
 ، en₁₁₁ ، کم fetch
 en₁₁₁ کم اجرا می شود (T₁₁₁) (T_E) machine cycle نو مرحله اول 0 دارد (1)

الخط 2000H مول ← (S₁₁) bus
 الخط 3E_H = op code مول ← (S₁₁) bus
 و (T₃ = 10) [op code] (S₁₁) bus 280 (T_E)
 سیستم فرمان بار فرمان (J₁₁₁) کم خواهد شد

2001H S₁₁ bus 280 ، 2nd fetch cycle نو می شود (2)
 en₁₁₁ در زیر رکاردها (PC) ، address bus (S₁₁) را
 (S₁₁) → 202H (T_E)

second cycle ، fetch cycle نو می شود *
 [bus (high)] inactive و [M₁₁₁] نشان می شود

2nd cycle نو می شود (T₁) (S₁₁) bus ←
 machine cycle memory (3)
 memory read می شود که باز این را
 : en₁₁₁ RD و MREQ پرسید

data bus و 280 و T_E (S₁₁، U₁₁ و V₁₁) (4)
 bus را کم کنید (T₁) (S₁₁) bus ← (S₁₁) bus
 bus = input bus و 280 و T_E (S₁₁) bus ← (S₁₁) bus
 T₁ و ACT و ACI و en₁₁₁
 نهاد را کم کنید و T_E و en₁₁₁ را کم کنید (5)
 CPU یعنی = machine cycle ← T_E

کامپیوٹر کا ایسا نامہ جسے داری دار ہے اس کا نام Wait ہے اس کا معنی ہے اس کا داری دار کو انتظار کرنے کا کام کرنا۔

SESSION 5

بے نہ ایجاد حافظہ ایجاد کرنے کا memory write machine cycle

Instruction o LD (HL), A

کامپیوٹر کو ایجاد کرنے کے لئے AC کا content کو کامپیوٹر کے bus پر پڑھا دیا جائے گا۔

اول یا Fetch opCode پر (ور) رکھ لیں (فولوو کریں)

وہ تکمیل کرنے کے لئے bus پر PC (280) کو تکمیل کرنے کا کام کرے گا۔

اس کا مطلب (ستارہ (لند تولی خانہ)) ہے MWP

وہی دوسرے کام کے لئے VVH کو کام کرے گا۔

وہی دوسرے کام کے لئے MREQ, RD کو کام کرے گا۔

2806 (T4 T3) کو opCode کو fetch کرنے کا کام ہے ②

memory write کا کام کرنے کا کام ہے MWP، اس کو fetch کیا جائے گا۔

III. VVH

Memory write (Free machine cycle) ③

bus پر داری دار کو HL کو داری دار کر کر کر تکمیل کرنے کا کام ہے۔

out port MREQ memory request

data bus (جہاں کو AC جیسے ایک داروں کا بروپریو

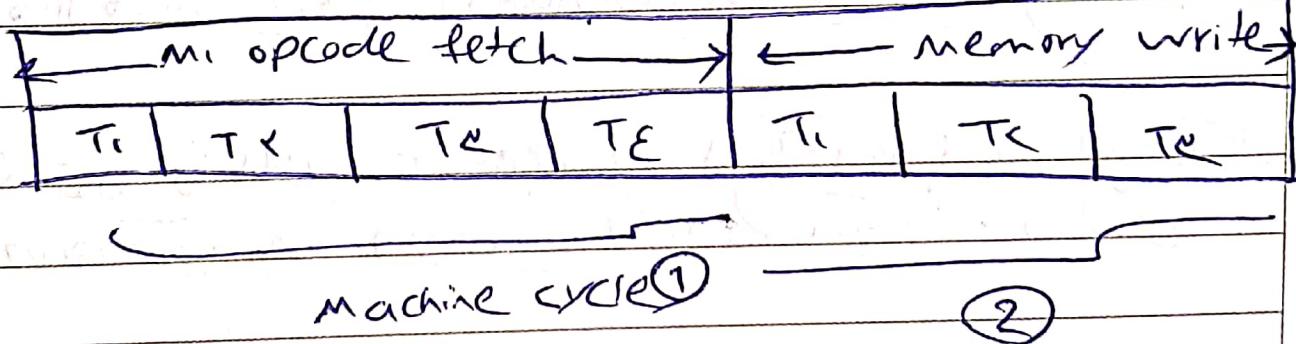
(MREQ کی وجہ پر) (T state) \rightarrow bus \rightarrow چھپا ④
bus line

(WR) 6 280 \leftarrow bus \rightarrow data bus (جہاں کو AC)

(data bus) 500 data byte 5 \leftarrow bus \rightarrow data bus (جہاں کو AC)
data bus کی وجہ پر WR, MREQ

help us \leftarrow bus \rightarrow T_E (جہاں پر وہ اڑا کر سکے گا) ⑤

high impedance \rightarrow data bus 6 \leftarrow bus \rightarrow data bus 6



device user program output
out instruction

memory address machine code

OUT (07H), A

2050	P3
2051	07

AC (جہاں کو AC)

data bus (جہاں کو AC)

0VH

memory contents

11010011	= P3H
00000111	= 07H

Ques.

Explain the machine cycle

Ans.

الخطوة الأولى هي fetch (استخراج) أو fetch of code (استخراج كود).

Second step is instruction fetch (استخراج命令) or machine cycle (خطوة الماكينة).

The machine cycle consists of three steps:

1. Addressing (العنوان) 2. Fetch (استخراج) 3. Execution (التنفيذ)

data bus (بيانات) و address bus (عنوان) و control bus (تحكم) يدخلون إلى CPU.

address bus (عنوان) يدخل إلى CPU (العنوان) من قبل مemory (الذاكرة) .

bus

data bus (بيانات) يدخل إلى CPU (العنوان) من قبل مemory (الذاكرة) .

WR

TORQ

YO

يتم إرسال إشارة (TR) (التحقق) من قبل CPU (العنوان) إلى مemory (الذاكرة) .

مُنْتَهِيَّةُ الْأَجْمَعِيَّةُ

3. Execution (التنفيذ) (الخطوة الثالثة) يدخل إلى CPU (العنوان) من قبل مmemory (الذاكرة) .

wait (انتظار) يدخل إلى CPU (العنوان) من قبل مmemory (الذاكرة) .

WR, TORQ & TR (STL) ← TR (التحقق) ← wait (انتظار) ←

(انتظار) ← wait (انتظار) ←

wait (انتظار)

ALE →
address latch enable

8085, وظائف

S1, S0, [10/M*]

status signal

ready (جاهز)

الخطوة الرابعة في الدورة المACHINE CYCLE (instruction) (الخطوة الرابعة في الدورة المACHINE CYCLE (instruction))

RD, P (أرجح)

memory

[10/M*] = 0 (جاهز)

[10] = 0 (جاهز)

[10/M*] = 1 (غير جاهز)

[10/M*]

S1

S0

machine cycle

0

0

1

memory write [MW]

0

1

0

~ read [MR]

0

1

1

opcode fetch [OF]

1

0

1

I/O write [IOW]

1

1

0

~ read [IOR]

1

1

1

interrupt acknowledge

[INA]

interrupt call (جاهز) INT (جاهز)

INTA, RD, WR

Machine cycle

RD*

WR*

INTA*

OF, MR, IOW

0 1

1

MW, IOW

1 0

1

Interrupt INTA

1 1

0

bus idle → BI

1 1

1

ATmega321

LDI - Local Immediate ①

constant \rightarrow writing

$L(D, R_d, K)$

$$15 \leq d \leq 21,$$

$$0 \leq k \leq \lfloor \frac{m}{2} \rfloor$$

MOV - copy register

این دلخواه کی از ریاست و مکمل وی زیرا توکی بی رید

(mov R0,RR)

$$Rd \leftarrow Rr$$

کی و ری، و کی و ری Rr

(figs 8, 9, 10 in Rr) — Rdt

১৫১

$$0 < r \leq r_1$$

ADD : Add without carry (3)

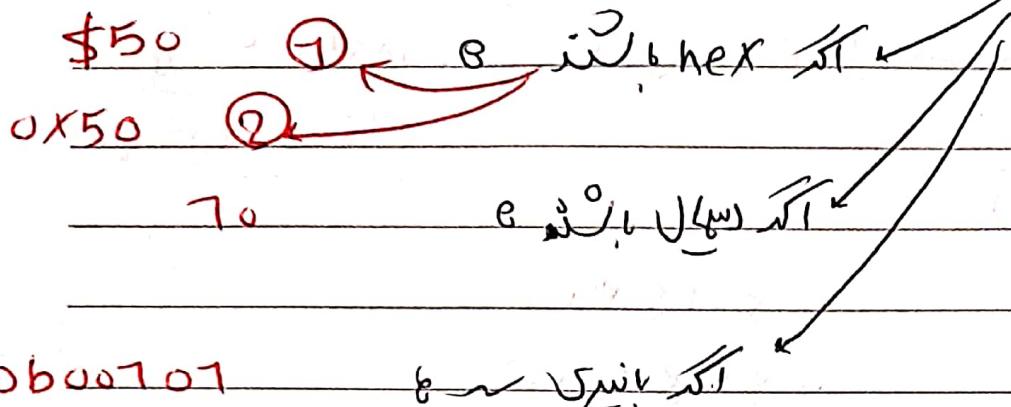
۲۷- لست، لیف (نقدگفت) فلک، باقم معزی کند و سقرا،
درینه لئو (عصر لوکی)

ADD Rd, Rr

$$Rd \leftarrow Rd + Rr$$

(?) ! (S15) Comment *

جواب



جواب

NO operation NOP ④

in 16 bits cycle

NOP

ADD with carry

جواب

ADC

⑤

$C \leftarrow R_d + R_r + C$

جواب

(ADC Rd, Rr)

$R_d \leftarrow R_d + R_r + C$

$0 \leq d \leq 1$

$0 \leq r \leq 1$

subtract without carry

SUB

⑥

جواب

SUB Rd, Rr

$R_d \leftarrow R_d - R_r$

$0 \leq R_d \leq 1$



subtract with carry

SBC

⑦

مکانیزم از کسر با بازگشایی و بازگشتی

(SBC Rd, Rr)

Operations

$$Rd \leftarrow Rd - Rr - C$$

(AND)

And

⑧

(AND Rd, Rr)

$$Rd \leftarrow Rd \cdot Rr$$

مکانیزم ریاضی AND

کسری غیر متعادل

logical or

OR

⑨

(OR Rd, Rr)

$$Rd \leftarrow Rd \vee Rr$$

مکانیزم ریاضی OR

کسری غیر متعادل

xor (Exclusive OR) (EOR)

⑩

(EOR Rd, Rr)

$$Rd \leftarrow Rd \oplus Rr$$



multiply unsigned

MUL

77

MUL Rd, Rr

R1 GR0 \leftarrow Rd \times Rr (unsigned \leftarrow unsigned \times unsigned)

8 bit \times 8 bit \rightarrow 16-bit

$0 \leq \text{Rd} \leq 1$

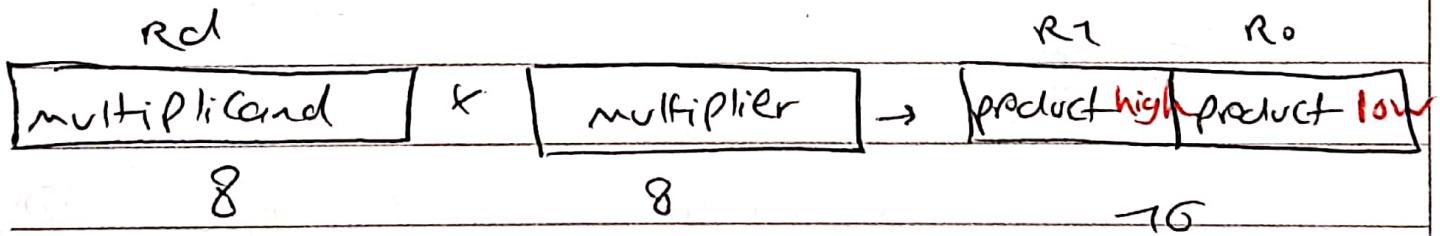
unsigned

R0 \leftarrow ~~Sign extension~~

R1 \leftarrow ~~Sign extension~~

~~8 bit \times R1 \leftarrow R0 \leftarrow Rr \times Rd~~

~~in \rightarrow sign extension will overwrite R1 \leftarrow R0 \leftarrow Rr \times Rd~~



multiply signed

MULS

78

MULS Rd, Rr

R1 GR0 \leftarrow Rd \times Rr (signed \leftarrow signed \times signed)

$-1 \leq \text{Rd} \leq 1$

~~no sign extension~~

$[2, c]$

~~(but no ≥ 0)~~

clear register

CLR (73)

(Clear کرنے کی) if clear ہے تو

(CLR Rd)

o ۰۰۱۱۰۱

اپنے رکھے رکھنے کے لئے XOR جسے مارے

Rd \leftarrow Rd \oplus Rd

(Clear کرنے کے لئے XOR جسے مارے

اپنے رکھنے کے لئے XOR جسے مارے

Increment

INC (74)

افٹافول کی سینے کے لئے [T] 8

(INC Rd)

Rd \leftarrow Rd + 1

o ۰۰۱۱۰۱

(Status register) SREG \rightarrow ۰ < \sim *

inc

Decrement

DEC (15)

کم کرنا ازٹے کی سینے کے لئے [T] 9

(DEC Rd)

o ۰۰۱۱۰۰

Rd \leftarrow Rd - 1

کم کرنا ازٹے کی سینے کے لئے [T] 9



one's complement

com 16

$\neg \text{num} \rightarrow (\neg \text{num}) \text{ (same)}$

com Rd

osclst

$$Rd \leftarrow \$FF - Rd$$

$\neg \text{num} \rightarrow 1111 1111 \rightarrow FF$

مثلاً $1111 + 1111 = 1000$, $1111 - 1111 = 0000$

1111 1111

$$\begin{array}{r} 0000 \\ - 0011 \\ \hline 1111 \end{array}$$

1111 0000

Up [] (GRE), C ~

[N & V] بوس ~

Up [] بوس ~

→ مبار

Status register: [] هو المقام في [R7] بـ Cilly ~ *

two's complement

NEG

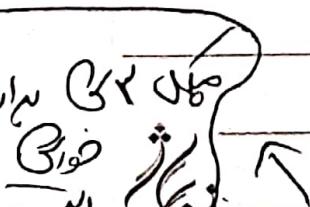
17

مثلاً $1111 + 1111 = 0000$, Rd

NEG Rd

$$Rd \leftarrow \$00 - Rd$$

مثلاً $1111 + 1111 = 0000$



$$\begin{array}{r} 0000 \\ - 0000 \\ \hline 1111 \end{array}$$

مثلاً $1111 + 1111 = 0000$

مثلاً $1111 + 1111 = 0000$

Rotate Left through carry ROL 18

(ROL Rd)



گام سیت (6) روتار را از اینجا بروز کر کنیم

دستور (rotate left) میں چه سبقتی کی دو نوعیں هیں

صعودی (Circular left shift) و دوسری (Circular right shift)

Rotate Right through carry ROR 19

(ROR Rd)



دوسری روتار را از اینجا بروز کنیم

روزی روتار

دستور (Circular right shift) میں چه سبقتی کی دو نوعیں هیں

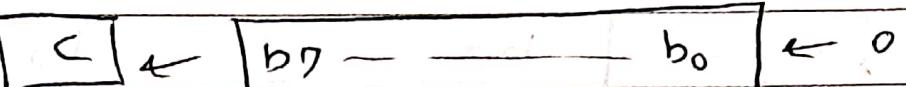
N معنی نویسندگان RV

Logical shift left

LSL (20)

LSL Rd

osd ۲۴



عملیات نویسندگان (کم از بیش) را درینجا بروزهای داشتند. ابتدا می خواستند بازدید کنند و سپس آنها را پاک کنند. این کار را با استفاده از دستور LSL انجام دادند. این دستور دارای دو فرمت است:

LSL Rd

LSL SREG

این کار را بازدید کنند و سپس آنها را پاک کنند. این کار را با استفاده از دستور LSR انجام دادند. این دستور دارای دو فرمت است:

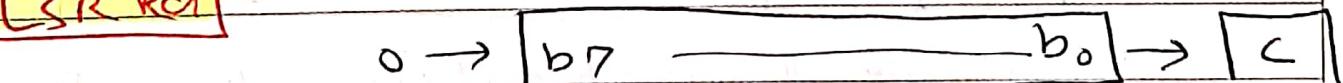
LSR Rd

LSR SREG

Logical shift right

LSR (21)

LSR Rd



این کار را بازدید کنند و سپس آنها را پاک کنند.

SREG

osd ۲۵

این کار را بازدید کنند و سپس آنها را پاک کنند.

کامپیوچر

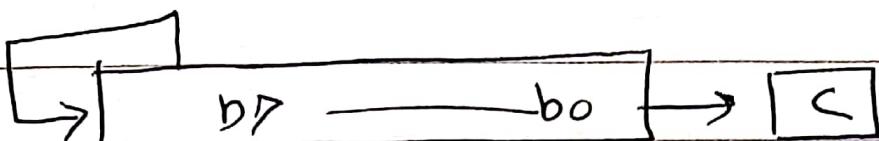
این کار را بازدید کنند و سپس آنها را پاک کنند.

این کار را بازدید کنند و سپس آنها را پاک کنند.

Arithmetic shift right

ASR

(22)



ASR Rd1

o Rd1 < 31

و این دستور را باید در راستای ریزی مانند جایگزین کرد
که از آن دستور ASR است که هم صحن و قوه دارد
و نیز SREG را درست

این دستور را ایداره کردن چند قسم می کنند این اسناد
در اینجا نشان داده شده اند

swap nibbles

SWAP

(23)

که این دستور یک nibble را بین دو nibble

SWAP Rd1

R1,16

R1,8

o Rd1 < 31

R(764) \leftarrow Rd1 (360)

R(360) \leftarrow Rd1 (764)

R1,16

R1,8

Branches (jump) ①

AVR \rightarrow jump targets

jump

CALL ②

to subroutine location

اے دلخواہ تر، اسی کیم بے محدود ریت
کیم کیم تر، اسی کیم بے محدود ریت

loop

Backs

counter

sum

$Z = 1 \leftarrow R_n = 0 \text{ if DEC } R_n$ رکھ رکھ رکھ رکھ
J.P BRNE Back

BRNE Back $Z = 0 \leftarrow$
J.P

Counter(SUM)

LDI R16, 10 ; $R16 = 10$ 6 J.P

LDI R20, 0 ; $R20 = 0$

LDI R21, 3 ; $R21 = 3$

$R20 = \text{sum}$

AGAIN ADD R20, R21 ; $R20 \leftarrow R20 + 1, 03$

DEC R16 (رکھ رکھ R16) counter

BRNE AGAIN Count = 0 کیوں تکمیل کریں

رکھیں

-N, counter

(N)

branch if not equal

BRNE

(24)

(BRNE K)

relative Δ to PC =

$\Delta = Z = 0 \rightarrow \text{if } Z=0, \Delta \text{ is zero}$
 $\Delta = Z = 1 \rightarrow \text{if } Z=1, \Delta \text{ is non-zero}$

$PC \leftarrow PC + \Delta + 1$

$\rightarrow \text{if } Z=0 \rightarrow PC \leftarrow PC + 1 \rightarrow Z=1 \rightarrow$
 $\Delta = Z = 1$

if $Rd \neq Rr (Z=0)$ then $PC \leftarrow PC + \Delta + 1$
else $PC \leftarrow PC + 1$

$-64 \leq \Delta \leq +63$

→ Slave *

LDI R16, 0x55 ; $R16 = 0x55$

OUT PORTB, R16 ; $PORTB = 0x55$

LDI R20, 10 ; load 10 into R20 outer loop count

loop_1: LDI R21, 70 ; load 70 into R21 inner loop count

loop_2: BCLM R16 ; complement R16

OUT PORTB, R16 ; load portB SFR with the

DEC R21 (inner loop) complemented value

BRNE loop_2

DEC R20 (outer loop)

BRNE loop_1 (first, last)

LDT R16, 0x55

three nested loops

OUT PORTB, R16

LDT R23, 10

LDP_3PLDI R22, 100 ③ ٣ لoops

LDP_2GLDI R21, 100

LDP_1RCOM R10 ٢ لoops

DEC R21

BRNE LOP_1

DEC R22

BRNE LOP_2

DEC R23

BRNE LOP_3

Branch if lower

BRLO ② ٥

(unsigned)

BRLO K

Rel SRR (ie 32 bits) - branch



PC \leftarrow PC + 1
C = 0

PC \leftarrow PC + 1
C = 0

-64 \leq K \leq +63

Now set CC

1111 1111

branch if same or higher

BRSH

(26)

BRSH K

if $R_{d1} > R_r$ (C=0) then
PC \leftarrow PC + K + 1

else PC \leftarrow PC

SA SKS + K3

if True (C=1)

if $R_{d1} > R_r$ (C=0) then PC \leftarrow PC + K + 1

else PC \leftarrow PC + 1

branch if Equal

BREA

(27)

BREA K

if $R_{d1} = R_r$ (C=1) then
PC \leftarrow PC + K + 1

else PC \leftarrow PC

SA K (K3)

if True (C=1)

if $R_{d1} = R_r$ (Z=1) then PC \leftarrow PC + K + 1

else PC \leftarrow PC + 1

branch if minus

BRMI

(28)

BRMI K

if $N=1$ then PC \leftarrow PC + K + 1

else PC \leftarrow PC + 1



branch if plus

BRPL

(29)

(BRPL K)

if $\oplus_{j=1}^K$ y_j is clear on \sim $\overbrace{N=0}$

if $N=0$ then $PC \leftarrow PC + K + 1$

else $PC \leftarrow PC + 1$

branch if overflow

BRVC

(30)

(BRVC K)

cleared

if $\oplus_{j=1}^K$ $y_j = 0$

if $\oplus_{j=1}^K$ y_j clear

if $v=0$ then $PC \leftarrow PC + K + 1$

else $PC \leftarrow PC + 1$

branch if overflow set

BRVS

(31)

(BRVS K)

if $v=1$ then $PC \leftarrow PC + K + 1$ else

$PC \leftarrow PC + 1$

into short jump

or into branch (slip *

long jump into (far branch) if

*

jump

Jump (31)

AVR unconditional branches

Equivalent needs $(SJT + \text{Time}) \leftarrow$

jmp ① ✓

Rjmp ②

Ijmp ③

jmp

jmp

(32)

Jump \rightarrow jmp

(jmp k)

$SK \leq EM$

$PC \leftarrow K$

relative jump

Rjmp (33)

(rjmp k)

jmp \rightarrow rjmp

$PC + 2K \rightarrow PC - 2K + 7$

indirect jump

jmp (34)

$PC(15\% \rightarrow 15\%)$

$PC(21810) \leftarrow 0$

Load direct from
data space

LDS

(35)

LDS Rd, K

جواب داده از تابع
برگشت

(ie) $Rd \leftarrow K$

$0 \leq d \leq 31$

$0 \leq k \leq 4095$

$PC \leftarrow PC + 2$

LDS R0, 0x300

جواب

R0 (جواب) 0x300, location (جواب)

LDS R1, 0x302

R1 (جواب) 0x302, location (جواب)

Store direct to data space

STS

(36)

داتا را در مکانی داده از تابع

STS K, Rr

$(K) \leftarrow Rr$

$0 \leq r \leq 31 \rightarrow 0 \leq k \leq 4095$

$PC \leftarrow PC + 2$

Load an I/O location to

IW

(37)

register

IW Rd, A

$Rd \leftarrow I/O(A)$

رال رسماً \rightarrow Rd میل جواب را از تابع I/O بگیرد

$0 \leq d \leq 31, 0 \leq A \leq 03$

$PC \leftarrow PC + 2$

IN R12, PIND

جاء من

R12 چیزی را PIND که local

و PINB کو PIND کیم کیفیتی داشت

IN R1, PIND

IN R2, PINB

ADD R1, R2

STS 0x300 > R1

 └───┘

location چیزی که (S0), R1

کوچکتر [\$300]

Store register to I/O location OUT

38

OUT A, Rr

6 Rr چیزی که داشت

و Rr کو

I/O (A) ← Rr

(S0) کو

I/O چیزی که register file

PC + PC + 7

پس AVR کو memory space چیزی که داشت

کوچکتر، چونکه code memory space

1K

Data memory space

کوچکتر، و داشت

Data memory space

(6 general purpose register) to GPR ①

32 x 8 register

status register ②

timer

I/O memory

I/O

I/O ports, serial communication

ADC

internal data SRAM ③

load direct from

LDS

Data space

innermost data space is L, w, b
instruction

LDS Rd, K

Rd \leftarrow (K)

PC \leftarrow PC + 2

0 ≤ d ≤ 31,

0 ≤ K ≤ 4096

LDS R0, 0x300

0 جلو

(جاءت), 0x300, جيت (جاء)

R0 جاء,

LDS R1, 0x302

R1 جاء, 0x302 (جيت جاء)

(store direct to data space) STS ④

STS K, Rr

(Store into memory if in data space)

(K) \leftarrow Rr

o SRSR, OSKS RCDW

PC + PC + 2

0,0,0,0,0,0 I/O memory \rightarrow location 10 *

I/O address ①

I/O registers

Data memory ②
address

(load an I/O location to register) IN ④

IN Rd, A

IN IN(Sig) 6 I/O Sig if
register file \rightarrow mem

Rd \leftarrow I/O(A)

o SASRN

PC + PC + 1

o SDRRN

R19 \leftarrow IN(Sig) [PIND C16] CPU

IN R19, PIND

Sig *

Location (I/O Bus), PIND, PIND C16 &
10x300

from data memory \rightarrow

STS (ie)

IN R1, PIND

IN R2, PINB ; load R2 with PINB

ADD R1, R2 ; R1 = R1 + R2

STS 0X300, R1

LDS IN نفاوت

IN IN IN

①

②

③

يكتب (كى) العالى (فينا) يرس IN
يكتب (كى) العالى (فينا) IN if (جى)
يكتب (كى) العالى (فينا) if (جى) لـ I/O register

الى (لـ AVR) يرس IN ④

الى (لـ AVR) يرس LDS 10

(Store register to I/O location)

OUT ④

OUT A, Rr

رس A رر دار

I/O space \rightarrow register file

I/O (A) \leftarrow Rr

$$0 \leq A \leq 99$$

$$0 \leq r \leq 1$$

program ROM space

(program counter)

pc

in کی درسته ای داده (جتنی) را

pc width

کاری که از پردازنده برای این کاری استفاده می شود

کنم develop را بتوانیم

Assembler Directives

• EQU ①

in کی تعریفی constant value

in کی تعریفی data item

label کی تعریفی constant

• EQU COUNT = 0X25

(LD)

R27, COUNT

⇒ R27 = 0X25

R27 (سازنده COUNT را)

R27 LDS را نمی خواهد → این از اینکه LDS نمی خواهد

جوابی بگیر و این را COUNT نویس

in کی تعریفی constant را

• SET ②

این از اینکه بزرگتر از EQU است

برای این کار

• ORG ③

(ابداً لـ دعـ نـيـاـ) (نـيـاـ دـعـ نـيـاـ) (نـيـاـ دـعـ نـيـاـ)

• INCLUDE ④

مـ قـاـيـلـ رـاـبـتـ بـرـنـاـجـ اـفـاـقـ مـ حـكـمـ

• DB ⑤

byte size (جـوـدـيـاـ) → (نـيـاـ دـعـ نـيـاـ) program memory

• DB oxb0101 binary

• DB 28 decimal

• DB 0XA hex

• DB 'S' ASCII

word size (جـوـدـيـاـ) program memory • DW ⑥
نـيـاـ دـعـ نـيـاـ

• ESEG ⑦

EEPROM

ESEG

• DB ab0101

• DB OXE

پـرـمـيـةـ (Code memory) سـعـيـاـ

• CSEG ⑧

CSEG

• DB ab7070

• DB OXE



SRAM

SRAM (جَوَابِيَّةٌ) .DSEG (9)

.DSEG

.DB ab701

.DB oxE

Buy (جَوَابِيَّةٌ)

.Exit (10)

.EQU SUM = 0x300 ; SRAM loc \$300 for SUM

.ORG \$00 ; starts at address 0

LDI R16, 0x25

LDI R17, \$34

LDI R18, 0b00110001 اُنْصَارِيَّةٌ

ADD R16, R17 R18

ADD R16, R18

LDI R17, 11 ; R17 = 0x0B

ADD R16, R17

STS SUM, R16 ; → SUM (جَوَابِيَّةٌ R16) مُفَارِقٌ
\$300 جَوَابِيَّةٌ

HERE jmp HERE ; عَلَى الْمُلْكِيَّةِ

EEPROM (جَوَابِيَّةٌ)

→ جَوَابِيَّةٌ 100 (جَوَابِيَّةٌ SRAM, w)

جَوَابِيَّةٌ 100 (جَوَابِيَّةٌ EEPROM)

AVR \rightarrow EEPROM (جواب)

EEPROM control Register

EECR

①

Data

EEDR

②

10 register
memory (جواب)

EEPROM Address register

EEARH & ③

EEARL

EEPROM

high - low

EEPROM (جواب)

(EEPROM Data Register)

EDDR

EDDR (جواب) \rightarrow EEPROM (جواب) (جواب)

(جواب)

(جواب)

EEPROM

جواب EEPROM (جواب) EEPROM is also called (جواب)

EEARH & EEARL

is like (جواب) (جواب) (جواب)

EEPROM \rightarrow location \rightarrow (جواب) (جواب)

(EEPROM control

EECR

register)



start

Read
write

جواب (جواب)
جواب (جواب)

EEPROM Read Enable \leftarrow EERE

EEPROM write Enable \leftarrow EEWE

EEPROM \rightarrow (ج)

مداخل و مخرج

عندما EEWE يكتب إلى 1 (ج) ①

عندها EEAR يكتب إلى EEPROM (ج) ② (ج)

EEPR ~ ~ ~ ~ ~ ~ ③

(EECR ج) يكتب إلى EEMWE ، = 1 ④

و EEMWE يكتب إلىcycle (ج) ⑤

EEWE = 1

ج

0x00FF location (ج) ، 'G' في AVR بيمثل عناوين

· INCLUDE "M16DEF.INC" EEPROM ;

WAIT G لانتهاء写操作 (ج) EEWE

SBIC EECR, EEWE

RJMP WAIT

EEARH (ج) ، (ج) - high byte

LDI R18, 0

LDI R17, 0XF

EEARL

~ low ~

OUT EEARH, R18

OUT EEARL, R17

EEPROM - data reg (ج) ، R16 (ج)

LDI R16, 'G'

OUT EEDR, R16

to one

SB1 EECR, EEWE ; set master write enable

SB1 EECR, EEWE ; set write enable to one

Bill ۱۰۰۵F ← EEP ۱۰۰۵F → ~~EEAR~~ AD

۱۰۰۵F
Bill

EEPROM is

address register

writes EEWE to location ۱۰۰۵F ①

(EEAR) job EEPROM must set ②

read enable ← EERE in ③

EEDR is EEPROM job ④

Data register

1) ۱۰۰۵F location to write in the PORTB (5th bit), is EEPROM
INCLUDE "M16DEF.WC"

LDI R16, 0xFF

OUT DPRB, R16 ; write EEWE
WAIT ; wait for write

SBI EECR, EEWE →

RJMP WAIT ; wait more

LDI R18, 0x50 ; load high byte of address to R18

LDI R17, 0x5F

OUT EEARH, R18 → ; write Bill into EEARH

OUT EEARL, R17 ; EEARL

SBI EECR, EERE ; set read enable

IN R16, EEDR ; read data to one

OUT PORTB, R16



out
R16 to
PORTB

load EEPROM data
register to R16

addressing modes

مذہبی ارکیو

ج) العنوان (لنك) بروتوكول data نحو cpu ←
addressing $\in \underline{W}$ بـ (لنك) بروتوكول
network mode

single register ① AVR، (پرکارا میں)

Register ②

direct ~~send~~

Register indirect(③)

Flash direct (A)

flash Indirect (5)

(immediate) addressing

single_register) ①

Two operands, one operand goes to the left

NEG R78; negative

(Com R19) : Complement

INC R20 ; increment

DEC R27 ; decrement

ROR R22 ; rotate right r22

register operand(s) را پس (پر) می‌خواهیم از هر چیزی که

— 1 —

R72 Original, 0x25

LPI R19 20X25

SUB1 R19, 0x6 subtract R19, 0x6

AND1 R1990b07000000 AND R1990X40

two registers)

در این صد اکری (۷۰) ۶ طبق حسن رایم کم داره هایی که در این بالعکس کارگنج

ADD R20, R23

SUB R29, R20

AND R16, R17

Mov R23, R19

DirectActa

addressing mode

Glossation (merges) a opdracht datter G, so (51, ✓)

آر سی ۱۰۰) سنا کیا ہے؟

1.1 RAM memory \rightarrow

اے آرک ریکووں بھی انریور →

النفادة في المدارك دارك كورس توفر رائسم

LDS R19, 0x50

R79 (Séjinsu)

memory loc

\$ 5

STS 0X40, R19

data (size: 1) 0x40_location (56) in 1, R19

data space

I/O Direct mode

(b) Standard I/O reg ~ See notes for details

on port

OUT, IN, I/O

IN R18, 0x16

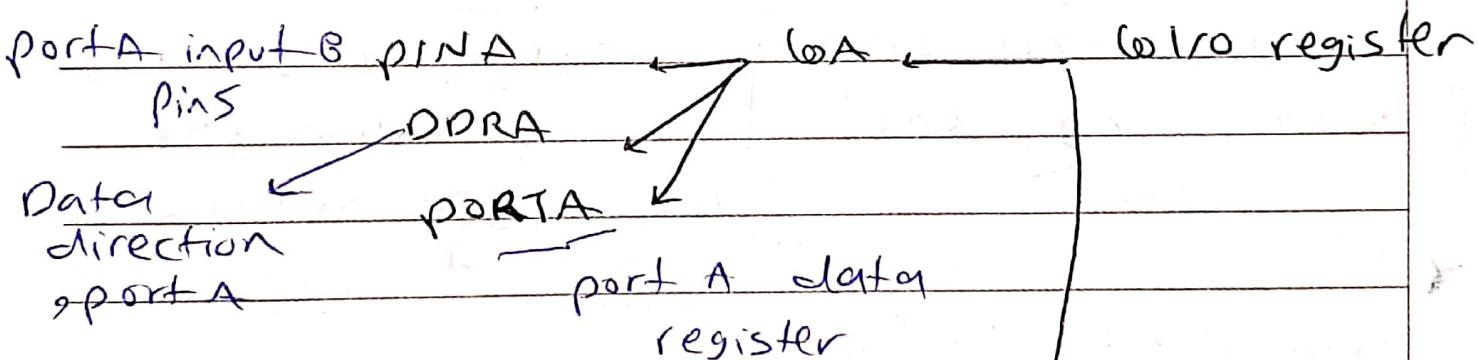
R18 = contents of location \$16

OUT 0x15, R18

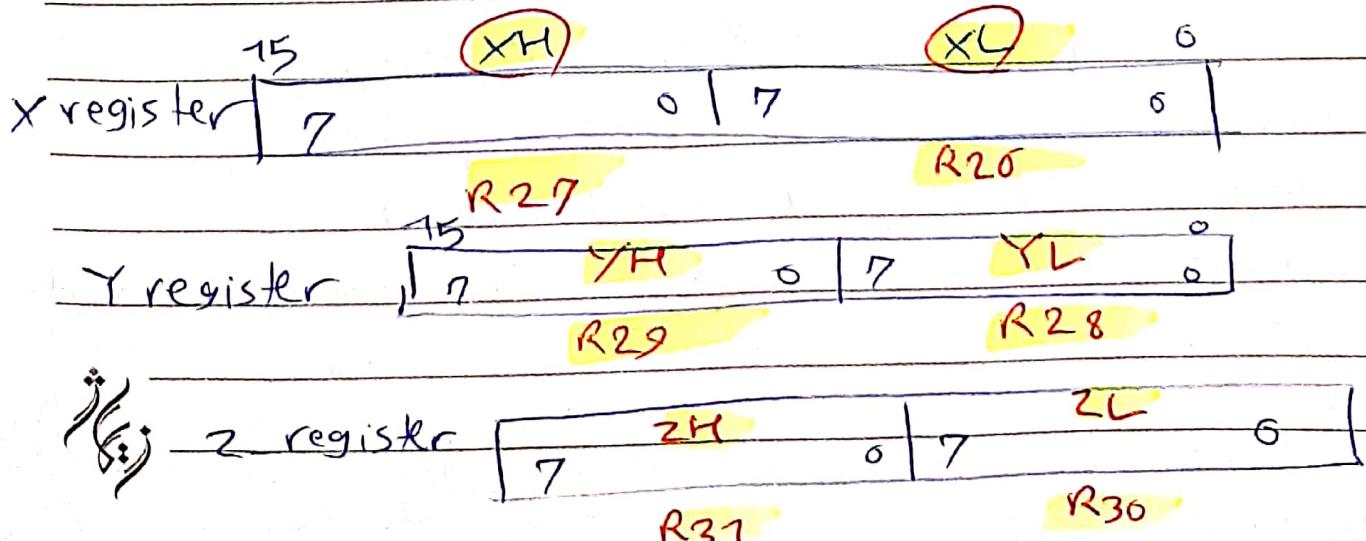
Pin 15, 0x15

R18

Port C (location \$15) = R18



Register indirect addressing mode

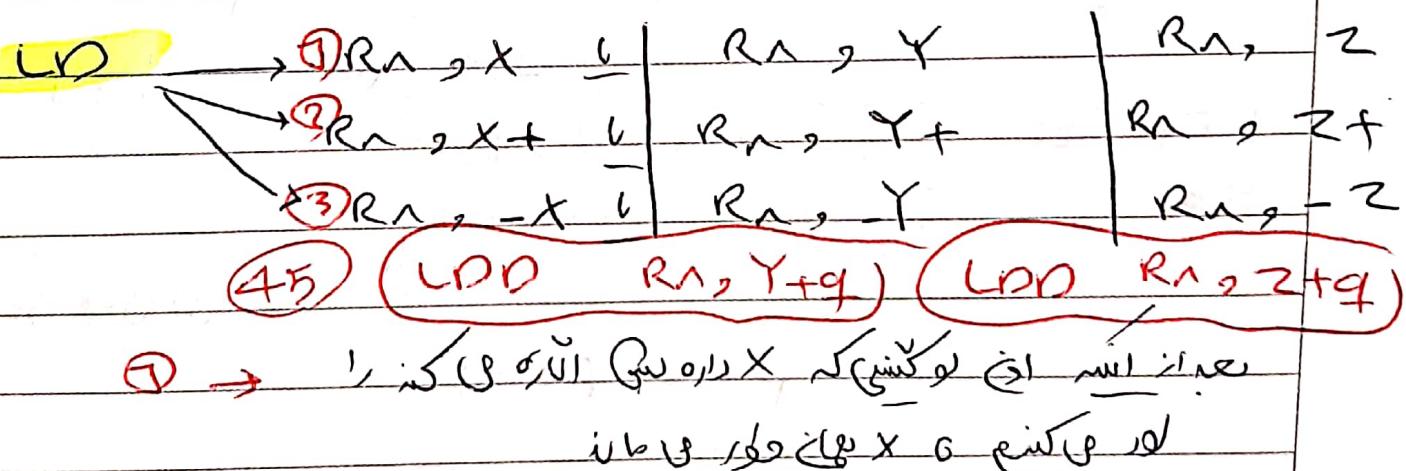


عنوان (عنوان) 62 \rightarrow X بـ pointer register دون

عنوان (عنوان) \$FFFF \rightarrow \$0000 از \rightarrow 61000
عنوان (عنوان) 1 (54464K عنوان)
کارکردن

دستور (INC ZL) از 51 pointer در کار کردن
عنوان (عنوان) میلیون کارکردن
RJL01

AVR auto Increment/Decrement of pointer
registers for LDD instruction



②

(Y increment) عنوان (عنوان) X

③ عنوان (عنوان) X کی location ای Rn که بـ X نوشتند اول
عنوان (عنوان) X کی location ای Rn



LDD R20, Z+5

R20 (عنوان, Z+5)

رسیسونز قرولز ۲ جم کے باسیں

رسیسونز ۶ جم، ۰۵ جم

(register indirect flash)

④ LPM R_{A, Z} load from program memory
رسیسونز ۱۸ جم، ۰۵ جم ایک ۲ جم کے خواندن

LPM R_{A, Z+}

Load from program memory
with post inc

Reads and increments Z

R_{A, Z}

R_d \leftarrow (Z) Z \leftarrow Z + 1

R_A \leftarrow Z

⑤ LPM رسیسونز ۶ جم، ۰۵ جم، ۰۵ جم LPM کی
رسیسونز ۰۵ جم، ۰۵ جم [R_d]

~~کوپریٹ~~

② LPM R_{d, Z}

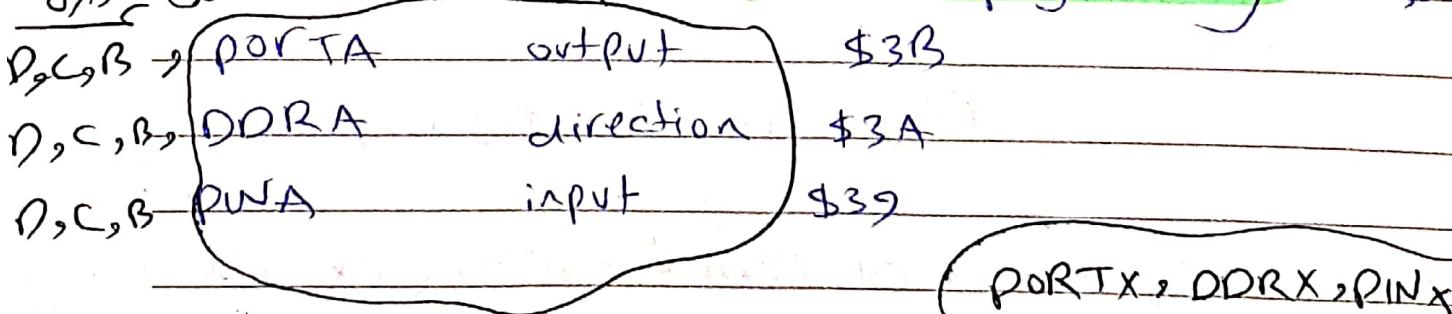
~~کوپریٹ~~

③ LPM R_{d, Z+}

گ بار ۰۵ جم، ۰۵ جم

PC + PC + 1

کوپریٹ
رسیسونز



output input, output port \leftarrow input
رسیسونز Port
DDR \leftarrow output, DDR \leftarrow input \leftarrow input

که این پین را PORTB گویند و این پین را output می‌نامیم

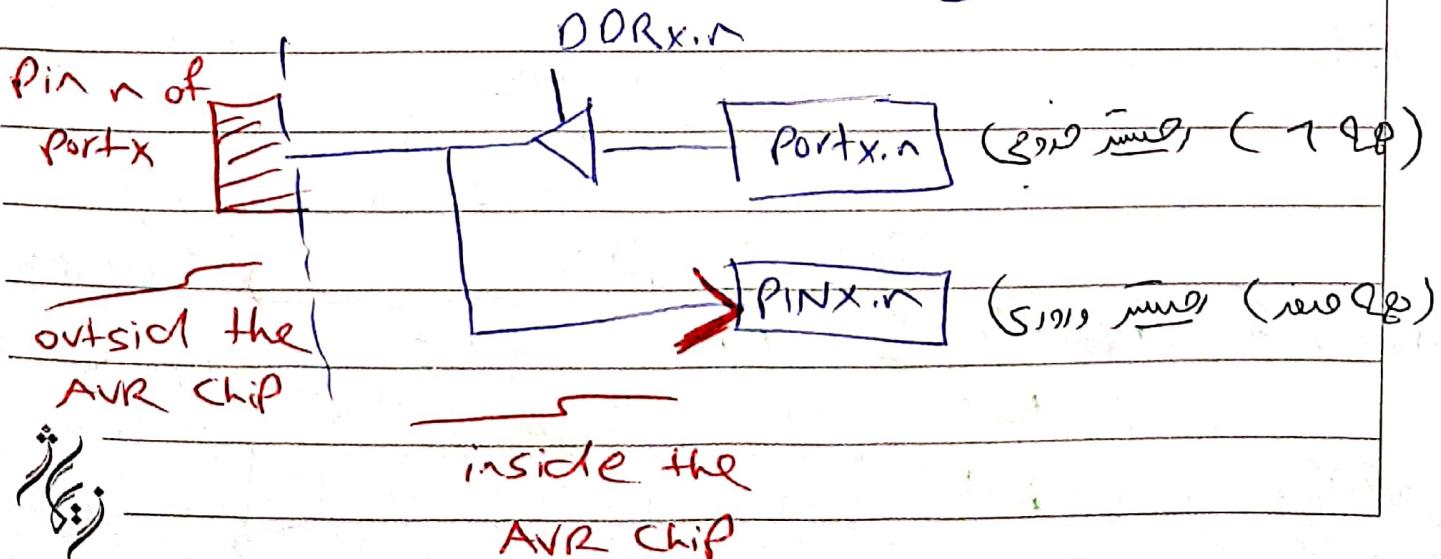
6 pin (SOP) DDRB چهار پین را بخواهیم [0b111111] output

DDRx چهار پین را بخواهیم OS با port که input می‌شود
که این پین را که کنترل درایور را که reach چهار port را داشت
و هر DDR چهار که input می‌شود port

مثال فردی که بول ندارد می‌تواند
(input بول شوند
output بول شوند)

DDRx را به port می‌کنیم ← reset با
که input را به port می‌کنیم . دو [0x00]

می‌توانیم ATMega32 را در چهار part می‌شوند
و هر چهار part یک output و input داشته باشد.



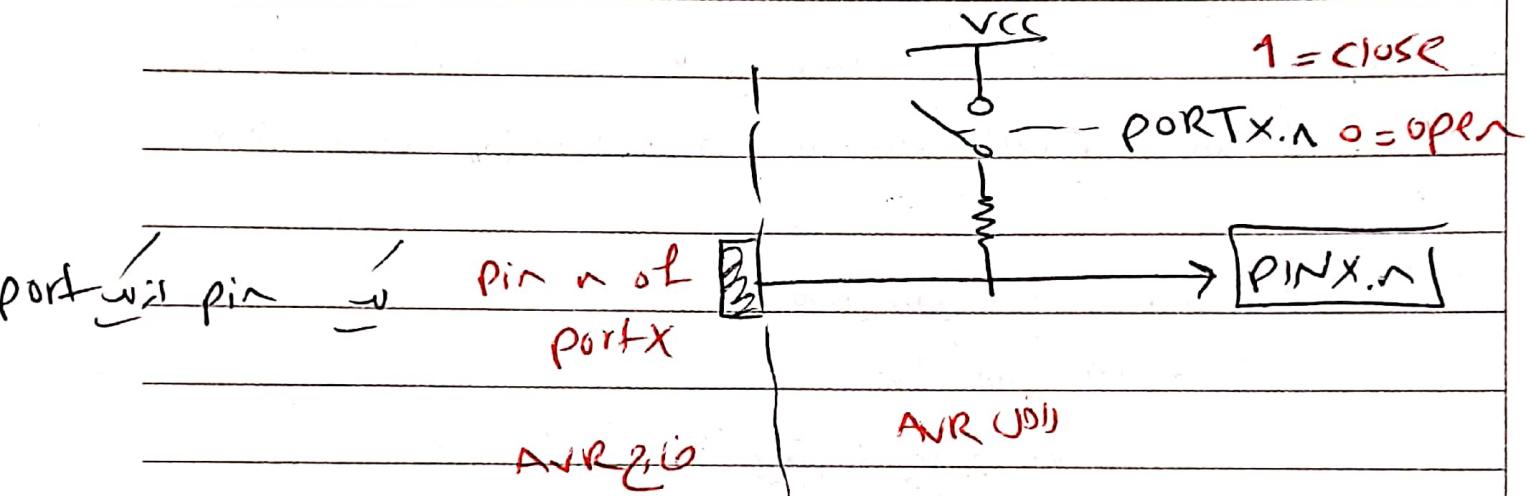
نحوه فوانی (pin) را کوئنچم

نحوه فوانی (PINX) را کوئنچم

نحوه فوانی

PORTX اینکه 6 OUT 6 pin را کوئنچم

العقاده



AVR در pin هر کسی را کوئنچم (pull up)

pull up معاوست هایی را کوئنچم (PORTX) اگر دلخواه باشد (pin) را کوئنچم (pull up) معاوست هایی و معاوست هایی معاوست هایی (pull up) اگر دلخواه باشد (pin) را کوئنچم (pull up) معاوست هایی و معاوست هایی (high impedance)

pull up معاوست هایی را کوئنچم (PORTX) (0) کوئنچم (1)

صفر

کوئنچم (1)

کوئنچم (0)

کوئنچم (0) کوئنچم (1) AVR

کوئنچم (1)

کوئنچم (0)

کوئنچم (1) کوئنچم (0) AVR

کوئنچم (0) کوئنچم (1) AVR

input & high

impedance

input & pull up

out 0

out 1

کوئنچم (0)

کوئنچم (1)

کوئنچم (0)

کوئنچم (1)

زیکی

کوئنچم (0) کوئنچم (1)

Reaching into memory
reach the entire byte ①
manipulate byte with logic ②
instructions

لے کر AVR کی فوائد اور ان کا عمل

set bits in register

SBR

AD

سے لہر کا سی سے صلیفن را (لوکی) رکھتے ہو تو اسکے لئے کون

SBR Rd, K)

Rd^o Rd v K

19 CDSCS

$$0 \leq k \leq r+1$$

~~Rd b , K w s or~~

Clear Bits in Register

CBR

48

is sp (spec) clear!, Red verb) → on (spec (SVO))

(BR, Rd, κ)

$$Rd \leftarrow Rd \cup (\$FF-K)$$

$$14 < d \leq 21$$

مکالمہ

$$0 < K \leq \omega_0$$

Bit store from Bit in register

BST

(4)

to T flag in SREG

status register T is temporary

BST Rd,b

$T \leftarrow Rd(b)$

$PC \leftarrow PC + 1$

$0 \leq b \leq V \rightarrow [0..b \leq V]$

Bit load from T flag in BLD

(5)

SREG to a bit in Register

bit in register b in SREG, T is temporary

BLD Rd,b

$Rd(b) \leftarrow T$

$0 \leq b \leq V$
 $0 \leq b \leq V$

skip if bit in register is

SBRC

(5)

cleared

SBRC Rr,b

skip if $Rr(b) = 0$

in P

in G + E + Z

if $Rr(b) = 0$ then $PC \leftarrow PC + 2 \text{ (or } 3\text{)}$ else

$0 \leq r \leq M$

$0 \leq b \leq V$

$PC \leftarrow PC + 1$

skip if bit in register is 0 PC \leftarrow PC + 1

skip if bit in register is 1 PC \leftarrow PC + 2
or (skip) PC \leftarrow PC + 3

skip if bit in register
is set

SBRS

(52)

SBRS rr, b (skip if bit in register is set)

if (b)

if RR(n) = 1 then PC \leftarrow PC + 2 (or 3)

else PC \leftarrow PC + 1

0 <= r <= 1

0 <= b <= V

set bit in I/O register SB1

(53)

SB1 (Set) to clear I/O register (Clear) (zero in)

SB1 A, b

Set A, b in

I/O(A, b) \leftarrow 1

SA <= V

0 <= b <= V

clear bit in I/O register

CB1

(54)

in (Pins), A \rightarrow 0, b \rightarrow 0

CB1 A, b

Set lower \rightarrow 0, b \rightarrow 0 in I/O

I/O(A, b) \leftarrow 0

0 <= A <= V

0 <= b <= V

skip if bit in V0 register
is cleared

SBIC

(55)

SBIC A, b

if $V_0(A \text{ and } b) = 0$ then PC + 2

(56) Register address

6-37

if $V_0(A, b) = 0$ then PC + PC + 2 (or 3)

else PC \leftarrow PC + 1

o $\leq A \leq ^n 1$

o $\leq b \leq ^n V$

skip if bit in V0 register SBIS
is set

(56)

SBIS A, b

if $V_0(A, b) = 1$ then PC + PC + 2 (or 3)

if $V_0(A, b) = 1$ then PC + PC + 2 (or 3)

else PC \leftarrow PC + 1

PC \leftarrow PC + 1 (for 1)

PC \leftarrow PC + 2 (for 2)

PC \leftarrow PC + 3 (for 3)

if $A > V$, then b



branch if bit in SREG is cleared

BRBC

(57)

جی یو بی، branch

BRBC S,K

جی یو بی و SREG (جی یو) نویسید
جی یو بی بی کشی کریں ← دلخواه نویسید
جی یو بی کشی کریں ← دلخواه نویسید

پنجه چوئے

PC - 63 (destination) PC + 64

جی یو

جی یو

جی یو بی کشی کریں ← دلخواه نویسید

if SREG(S) = 0 then PC + PC + K + 1

else PC + PC + 1

0 ≤ K ≤ 7

-4 ≤ K ≤ 4

branch if bit in SREG is set

BRBS

(58)

if SREG(S) = 1 then PC + PC + K + 1

BRBS S,K

status register, پس نویسید

جی یو بی کشی کریں ← دلخواه نویسید

PC + PC + K + 1

K

جی یو بی کشی کریں

PC + PC + 1

جی یو بی کشی کریں (پلک) دلخواه نویسید

Bit set in SREG

BSET

(59)

کسی، status register کو یہ نہ کسی

BSET S

0 < SSV

PC \leftarrow PC+1

SREG(S) \leftarrow 1

Bit clear in SREG

BCLR

(60)

کسی، SREG کو یہ نہ کسی

BCLR S

0 < SSV

PC \leftarrow PC+1

SREG(S) \leftarrow 0

SREG (Flags are Jioed)

action
SEC set carry C=1

action
CLC ~ carry C=0

SEZ set zero Z=1

CLZ ~ zero Z=0

SEN set negative N=1

CLN ~ negative N=0

SEV set overflow V=1

CLV ~ overflow V=0

SES set sign S=1

CLS ~ sign S=0

SEN set half carry H=1

CLH ~ half carry H=0

SET set temporary T=1

CLT ~ temporary T=0

SEI ~ interrupt I=1

CLI clear interrupt I=0

جاء في المقدمة أن GPR هي مemory RAM

وهي تسمى internal RAM

general purpose register

GPR

\$195 location لعنوان الذاكرة في المقدمة

internal RAM

Port B هو نوع من الذاكرة

Port B

location location Port B

0x195 ↑

EQU myreg = 0x195

LDI R16, 0xFF (ob1111 1111)

XX OUT DDRB, R16 XX

output port 1, portB

R16 myREG

AGAIN LDI R16, myREG

SBR S R16, 0 R16

RJMP OVER

CBR R16, ob00000001

skip over

OVER BUT PORTB, R16

PORTB

JMP AGAIN

RJMP

skip (جumper)

skip

CBR & clear bits in register R16-R16 (\$FF-K)

K = ob00000001

(Macro)

در راه یا گروه از دستورات که در این میان task برای یک هدف مشترک دارند را می‌توان خاص نام داد و آن را می‌توان مکرر کنن

مان نوشت که راکتیو کن ، تراویرها

MACRO name پسوند تعریف مکرو

ENDMACRO

پسوند تعریف مکرو

می‌شود، پسوند 10 بار از

برای اینها را با سیفی @

@0

و @9

معنی دارد که تعریف از 0 تا 9 باشد

هزارهاست کن

I/O register بینیم بین دستورات دستورات اینست و می‌شود ~~که~~
data RAM

macro loadIO

LDI R20, @1 بروامدوم را مینم

OUT @0, R20 R20 تهی

endmacro

برای اینست اول 1, R20

LoadIO PORTA, 0X20

PORTA(5,4,3,2,0,1,0X20)

ODRC (جُودِي) 1111 1111

(PORTB_C_OUTPUT)

② .EQU VAL_1 = 0xFF

Load10 ODRC, VAL_1

③ Load10 SPL, 0x55

SPL (جُودِي) 0x55

β include (جُودِي) بِلَغَةِ بَيْنَمَعْ وَ مَرْصُوقَ نَسَانِ دَالِيَّةِ مَكَارِدِ رَأَى فَاعِلِي بَيْنَمَعْ وَ مَرْصُوقَ نَسَانِ دَالِيَّةِ كَمْ

macro ! [PORTB_TOGGLE] *

.INCLUDE "M32DEF.INC"

.INCLUDE "MYMACRO1.MAC" ; get macro

اَنْ فَاعِلِي مَكَارِدِ

.ORG 0

Load10 DRRB, 0xFF → [PORTB_C_OUTPUT]

L16 Load10 PORTB, 0x55

DELAY R18, 0X70

Load10 PORTB, 0xAA

DELAY R18, 0X70

C JMP L1

Call subroutine (جُودِي) (S1 ←) (CALL)

Call task (جُودِي) (task ←) (to subroutine)

Call sub routine (جُودِي)

+ save S1, S2, S3, S4 ← (to sub)

Call save (جُودِي)



long call	CALL	①	↳ Call subroutine
recursive call	RCALL	②	
indirect call to Z	ICALL	③	
extended indirect	EICALL	④	
call to Z			

position to Z is fixed
but needs first ~

Long Call to a subroutine

CALL

in SP call to Subroutine → 6 bytes given by K (50)

stack (for return call) will be K

fixed

call uses in SP (post-decrement) to stack pointer

(CALL K) ① PC ← K devices with 10-bit PC

728KB program memory

maximum

② PC ← K devices with 12-bit PC,

8MB program memory maximum

① CALL K $0 \leq K \leq 64K$ PC ← K STACK ← PC + 2

SP ← SP - 2

(SP) \leftarrow SP - 2 (L = 24)

② CALL K $0 \leq K \leq 4M$ PC ← K

STACK ← PC + 2

(SP) \leftarrow SP - 2 (L = 24) SP ← SP - 3

(4K word(s) =
8KB)

relative call to subroutine

RCALL

(RCALL K)

PC - 2K + 1 (word) → S01 → (word) Call
to stack (S01) = S01. (PC + 2K) (word)

① PC ← PC + K + 1 devices with 16 bit PC → 128 KB
program memory max

② PC ← PC + K + 1 ~ ~ 22 bit PC → 8 MB ~ ~

RCALL K 2K ≤ K ≤ 2K PC ← PC + K + 1

STACK ← PC + 1

SP ← SP - 2

14 13 12

RCALL K 2K ≤ K ≤ 2K PC ← PC + K + 1

STACK ← PC + 1

SP ← SP - 3

15 14 13

Indirect call to subroutine

ICALL

Op (S01) 4M (S01) → in op call to subroutine
(words)

ICALL

Arg to stack (S01) ← (S01)

(.)(in operand)

(in post decrement, 6 SP)

① PC(1580) ← 2(1580) 6 16 bit PC

② PC(1580) ← 2(1580)
PC(21816) ← 0 6 22 bit PC

① ICALL d/i operand STACK \leftarrow PC+1

SP \leftarrow SP-2

(↓14 ↓15)

② ICALL STACK \leftarrow PC+1

SP \leftarrow SP-3

(↓13 ↓14)

extended indirect call to

EICALL

subroutine (↓15) \rightarrow 4M words (↓15) to indirect call

d/i (5, 8, 16)

EICALL

PC(15₁₀) \leftarrow Z(15₁₀)

PC(21, 675) \leftarrow EIND

(↓ END)

NO SPACE

EICALL d/i operand STACK \leftarrow PC+1

SP \leftarrow SP-3

(↓13 ↓14)

return from subroutine

RET

RET PC(15₁₀) \leftarrow STACK 8 19 bit PC

PC(21₁₀) \leftarrow STACK 8 22 bit PC

① RET d/i operand

SP \leftarrow SP+2

② RET ~ SP \leftarrow SP+3

اکسیوں stack کے ساتھ CPU کی تحریر کیا جائے

Stack pointer (SP) register

I/O space (جگہ تحریر، VR)

memory

SPL (low byte of SP)

SPH (high byte of SP)



stack (جگہ تحریر)

stack کے تکمیل کی جائے پر **PUSH** ①

اکسیوں stack (جگہ تحریر) کے SP

اکسیوں SP کے تحریر دار کر کر push کی جائے تو اس کے زیر پر

اکسیوں کے SP

اکسیوں کے SP

(PUSH Rr)

o srcc

PC \leftarrow PC + 1

SP \leftarrow SP - 1

CPU کے تحریر کر کر دینے کے لئے stack کے پر **POP** ②

اکسیوں کے SP

تھیں، اس کے تحریر کی کر کر دینے کے لئے top location

(POP Rd)

o srcc

PC + PC + 1 \leftarrow PC

SP \leftarrow SP + 1 \leftarrow stack

اکسیوں کے SP کو دینے کے لئے دینے کے لئے اس کے تحریر کر کر دینے کے لئے

جذب ایجاد نویسندگان higher memory ; stack
location

lower memory location

کد CPU می‌دهد (پرینت) یکی از آنها یکی از آنها

جذب (جذب RAMEND) ۶ AVR assembler (جذب
ویرایشی، RAM location
sp (جذب)، RAMEND)

stack ! RET ,CALL blets ۴
جذب CPU ۶ دستورات فرایند فرایند subroutine ۱

جذب stack CALL ، جذب

۷ return (جذب) از کجا برگشتی این دستور

PUSH (CALL) onto the stack

top locations (جذب) ۶ subroutine دستورات RET (جذب) ۲

PC (جذب) ۵ stack

SP (جذب) ۶ SP

POP (CALL) into

the PC

invok کرنے کی طرح کیا ہے مکروہا **Macro vs Subroutine**

10 بار 10 دلتوں کا مکروہ رکھوں 10 دلتوں کا مکروہ

فراہم کرنے کے لئے 100 دلتوں کا مکروہ رکھوں

Subroutine کا مکروہ

اگر 10 بار فراہم کرنے کا معنی ہے اندازہ رائی دلتوں

کا مکروہ

ایسی فراہم کرنے کا معنی

کے لئے RET کا کام کرے + کارکردن کے لئے function

ایسی اجنبی

اوپر اضافی کام کرنے کا معنی

ستارا

تعدادی

انتعارہ پر کیا stack (رہائی) کر مکروہ از فنا (to subroutine)

کیا کیا

AVR Timer Delay

چھوٹی تاخیر کرنے کے تکمیل کیا صفاور کے تکمیل کیا

کام کرنے AVR کا نامیں تاخیر کرنے کے تکمیل کیا

کام کرنے کے لئے تاخیر کرنے کا معنی

crystal frequency

(5MHz)

AVR design

کامپیوٹر کا ایک ایجاد کرنے والے دستورات کا مجموعہ ہے۔

Machine Cycle

Machine cycle کو 1 نیچے گئے ہے اور یہ بڑے ترے

Oscillator (فرکنڈ) کا Machine cycle ہے

Oscillator period نے 1 machine cycle کا

oscillator (فرکنڈ) کا 1 machine cycle کا

Instruction cycle is $1/8 \text{ MHz} = 0.125 \text{ ms}$ (microsecond)

$$\frac{1}{\text{ms}} = \frac{1}{0.125} = \frac{1}{12.5} \times 10^6$$

crystal frequency = 16MHz

$$16 \times 10^6 = 16 \text{ MHz}$$

$$4 \times 10^6 \text{ ms}$$

1MHz کی فرکنڈ کا AVR پر 10 جانے والے دستورات کا مجموعہ ہے۔

① LDI

1
to cycle time

$$1 \times 1 \text{ ms} = 1 \text{ ms}$$

6 cycle time $\times 10$ (جیسے ہے) =

60 جانے والے دستورات

② JMP

3
4

$$3 \times 1 \text{ ms} = 3 \text{ ms}$$
$$4 \times 1 \text{ ms} = 4 \text{ ms}$$

③ CALL

مکرری کیلیکسیکل گو 6 دیکٹیویٹریں

DEF

0 = (line)
0 \rightarrow 0

کامن، زین، دلیلیں یا گلہ
نیز تانڈر چکیا کی

DEF

COUNT=R20

16-255

لے جائیں

DELAY & LDI

COUNT \rightarrow 0xFF

AGAIN & NOP

1

NOP

1

DEC

COUNT

1

1 بیت سیکنڈ

BRNE

AGAIN

2/1

RET

LDI

$$(1 + (1+1+1+2) \times 255) - 1 + 1 \times 0.1\text{ms}$$

$$= 15 \times 10^{-4} \text{ s}$$

BRNE

1Kv/9ms

setting counter

①

Delay Subroutine

A loop ②

10ms - body 10ms delay گھونٹو



1MHz = (Instruction time + memory delay) \approx 62.5us

.ORG 0

LDI R16, HIGH(RAM^N)

OUT SPH, R16

LDI R16, LOW(RAM^E)

OUT SPL, R16

BACK B

LDI R16, 0X55 R16 Set to 0X55

OUT PORTB, R16

Subroutine Delay CALL DELAY

LDI R16, 0XAA

OUT PORTB, R16

CALL DELAY

RJMP BACK

Delay subroutine 0X300 subroutine

.ORG 0X300

255 = Counter

DELAY: LDI R20, 0XFF

(1)

AGAIN:

NOP

(2)

NOP

(3)

DEC R20

(4)

Delay B

(1+((1+1+1+1)*255))

-1+4)x

1ms =

15ms

NEARNE AGAIN ;

(5)

RET ; return to caller

(6)

branch (10us) to (10us)

if 1

(7)