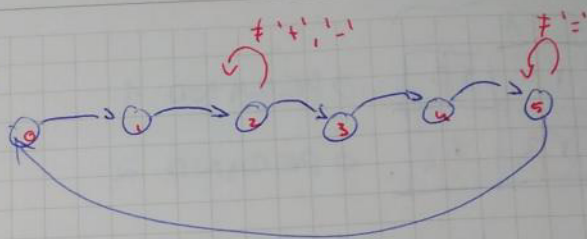


PRÀCTICA 7

little endian

$$\begin{array}{r}
 \boxed{10}_{11} \quad \boxed{253}_0 \\
 + \quad \boxed{3}_2 \quad \boxed{2}_3 \\
 \hline
 = 5
 \end{array}$$



codi ASCII '+' → 43
0x43

ASCII '-' → 45
0x45

amb el picocom no hi ha caràcters que tinguin un codi ASCII 0, per tant, hi hauria números que no podria entrar.
machine simple → coteran → internament se l'acaba.

Mag: complexa:

$$\begin{array}{|c|c|} \hline 13 & 10 \\ \hline 43 & 253 \\ \hline \end{array}$$

per enviar operadors hem d'enviar 4 lletres

SWAP

bis endian → problemes

$$\boxed{0000\ 0010}$$

$$\boxed{0000\ 1111}$$

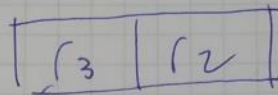
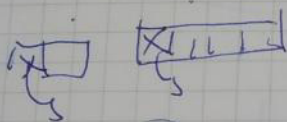
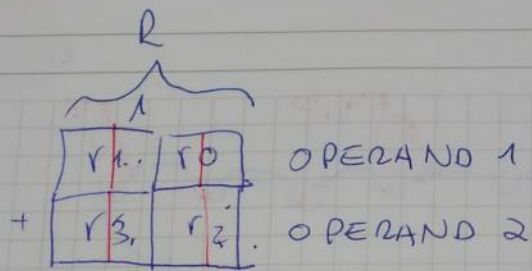
F → SWAP

$$\boxed{1111\ 0000}$$

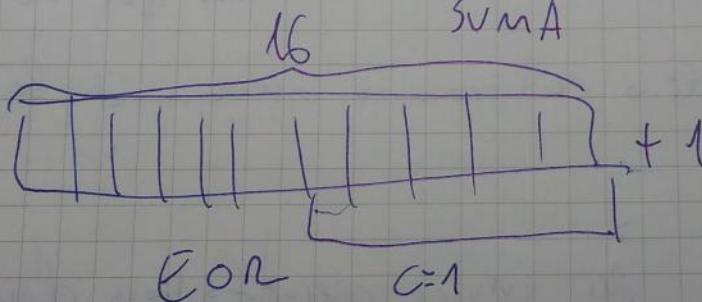
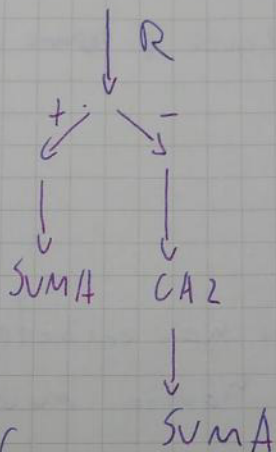
$$\text{OR } \boxed{0000\ 0010}$$

$$\boxed{1111\ 0010}$$

convertim F2 en binari



①



ADC Rr

Eor

Add +1

Eor

Adc +0 (+c)

MODES D'ADREÇAMENT

Registres propòsit general \rightarrow 0,32 (5 bits)

Memòria de clades \rightarrow 0,64K (16 bits)

CDS \rightarrow 2 opcodes (2 clades) per l'adreça de la RAM, 16 bits

INDIRECTAMENT \rightarrow



LD Rd, X

MODES D'ADREÇAMENT

a) Immediate

LPI R16, U U \rightarrow 8 bits

!!! 5 bits 16 bits
CDS r16, (U)

(U) \rightarrow 16 bits IN, OUT

ESPODMEN
CONSIDERAR
IGUALS

b) Directe

c) Registre

MOV r16, r0

d) Indirecte

LD r16, X \rightarrow POSICIÓ DE MEMÒRIA

etiqueta de l'adreça

dada = $\boxed{13}$

on X \rightarrow r27:r26

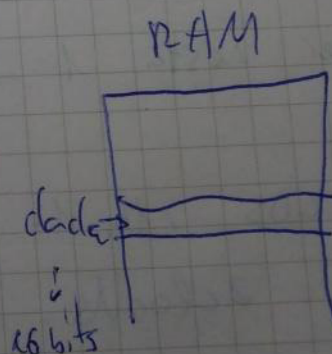
Y \rightarrow r29:r28

Z \rightarrow r31:r30

LPI r27, hi 8(dada) byte de més res

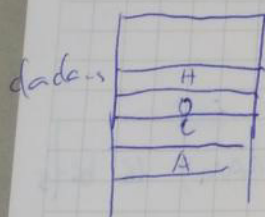
LPI r26, lo 8(dada) byte de menys res

LD r16, X



Amb això, si vols canviar l'adreça dada, només hauràs de canviar l'assignació $dada = \boxed{}$

si vull:



LD r16, X+

Posa a r16 el contingut de X després
X passa a valdre X+1 (dada +1)

Forcem

LD r16, X+

LD r17, X+

LD r18, X+

LD r19, X

LDI r27, hi8(dada)

LDI r28, bi8(dada)

tocna: LD r16, X+

CALL TX

rjmp tocna

Poc acabar:

tocna: r16, X+

CP r16, 0 \Rightarrow TST r16

BREQ surf

CALL tx

rjmp tocna

test for zero or
minus

$Rd \leftarrow Rd - Rd$

Activa Z si
son 0

surf: }

LD r16, X-

Primer resta 1 a X i després el corregeix X (ja modificada)

LDS r16, 17 \Leftrightarrow MOV r16, r17

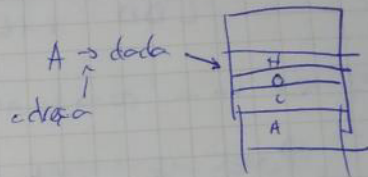
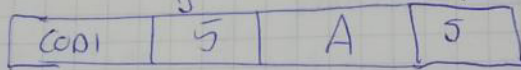
2 clocks

1 clock

e) Indexat **L'AVR NO EL TÈ**

MOV r16, A(r12)

opcode



a r16 poso:

A(0) = H

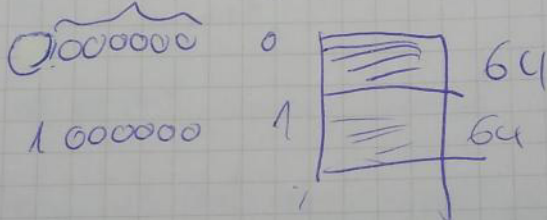
A(1) = O

A(2) = L

A té la constant de l'etiqueta dada

suro 12 a la constant A

No els posos

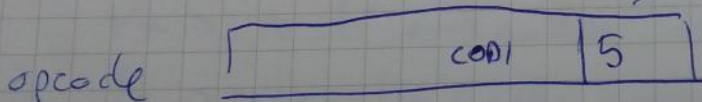


8) Base + Index **L'AVR NO EL TÈ**

MOV r16, (r3+r5)

és com r3(r5)

9) stack **L'AVR NO EL TÈ** registre



Carrega el valor de la pila

No existeix però seria PMOV r16

poso a r16 el contingut de l'adreça de la stack pointer

LDD r16, Y+q

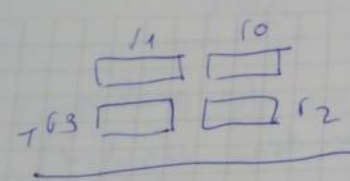
LDD r16, Z+q

q és una constant, és un índex

Posa a r16 el que hi ha a Y+q, sense canviar Y

opcode

C	5	7	9
---	---	---	---



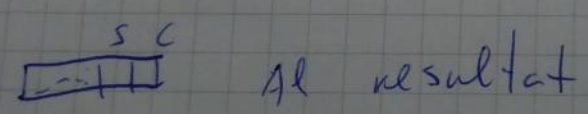
suma: ADD r0, r2
ADC r1, r3 $\xrightarrow{+ \text{ carry?}}$
ret

si estem sent una suma, el carry de la suma amb doble precisió és el carry global.

si som resta, el carry global de la suma cal negarlo ja que és el borrow

RESTA: $C_n(r3:r2) \rightarrow C_n(r3:r2)$ $r6=0$
coll suma Com r3
Caldria $*C = \bar{C} \rightarrow$ el nega per convertir neg r2
nica / el carry en borrow. $*C = \bar{C}$
Com calcula ret Adc r3, r6
el flag si. Al ser la resta vull borrow.
RET
Com a complement a 1 \rightarrow inverteix 1 i 0
(sense sumar 1)

* al ser neg, el carry no és carry, és borrow



Resultat binari \rightarrow 6 caràcters

en la resta representa un borrow
resultat (hex)

el flag s'ha també queda calculat automàtic

'0' '0'
'1' '1' FA 10

'5' '0' 'C' '0' ' ' ' ' Resultat hexa
1 1 FA 10
0 \rightarrow +
1 \rightarrow -

000 116, 2+9
 7 é 5 um ...

el primer 8 → ^{neg} 1000

neg. 1

$$\begin{array}{r} 00 \quad 01 \\ + \quad 6FF \quad FF \\ \hline [1] \quad 780 \quad 00 \end{array}$$

N = 1 Soma dos positivos i
 V = 1 → em dona negativ!

$$S = N \oplus V \rightarrow S = 0$$

$$\begin{aligned} N &= 0 \\ V &= 0 \\ S &= 0 \end{aligned}$$

Problem sumar i que S = 1?

Si: $0 + (-1) = -1$ $S = 1$

$$\begin{array}{r} 00 \quad 00 \quad (0) \\ + \quad FF \quad FF \quad (-1) \\ \hline FF \quad FF \end{array}$$

$$\begin{aligned} N &= 1 \\ V &= 0 \\ S &= 1 \end{aligned}$$

Si Soma positiv i
 negativ, mai dona
 over flow!!!

$$\begin{array}{r} 0001 \\ + 7FFF \\ \hline 8000 \end{array}$$

a) sense signe \rightarrow l'únic que em procés és calcular el carry
 $C=0$

b) Amb signe

$$V=1 \quad N=1 \rightarrow S = N \oplus V = \boxed{0}$$

$\boxed{0} \ 8000$

$$\begin{array}{r} 0001 \\ + 8FFF \\ \hline 9000 \end{array}$$

a) Sense signe

Si treballen sense signe, la N perd significat, no existeix

$$C=0$$

b) Amb signe

$$V=0$$

$$\begin{array}{r} (+) \\ + (-) \\ \hline \end{array} \quad \begin{array}{l} V=0 \\ \swarrow 1 \end{array}$$

CA2

$$\begin{array}{l} 1000 \quad F \quad F \quad F \\ 8FFF \rightarrow 0111 \quad 0000 \quad 0000 \quad 0001 \rightarrow -28673 \\ 9000 \rightarrow -28672 \end{array}$$

$$N=1$$

$$S = \boxed{1}$$

$$\boxed{1} \ 9000$$

\hookrightarrow és negatiu!

250

al revés

→ al revés
Sumas dos negativos, dána positivo

→ soma dos positivos é obtida
negativa

↑ datos
↓ programas

Section date

tot el que unsui era
unic a memòria de
la des

*include "taula 8.5" per treballar
amb altres sistemes

anirò posant

1 byte = 8 bits

codi asci de codu

$$b + c$$

al final, després
de la última lletra,
posa un zero ~~hi~~ valor,
no hi ha cap caràcter de
codi ascii que valgui 0.

section - bss

vesades:

a l'adreça vesades hi posa la
dada 0

la següent
posició de
dades

-byte 0

-space 32,0

escriu 0 a 32 posicions diferents
a partir de buffer

• -global -- do-copy -- data

* -global -- do-clear -- bss

taula cl:

A
B
C
D
...
Z
x
y
...

taula Z: taula cl - bin

sen la mateixa adreça

* -bss → des variables que hi definim, és a dir tot el
que hi definim, en el moment de reset valdran
zero.

• -data → al fer un reset, defineix allà

Està a la memòria de programari, al fer

Startfiles → un reset les passa a la de programari.

Al començar, després del reset, les carrega a
data

$$\begin{array}{r} 0010 \quad 2 \\ 0100 \quad -4 \end{array}$$

$$1011$$

\rightarrow

$$\begin{array}{r} 0010 \\ \underline{+1100} \\ 1110 \end{array}$$

$$C=0$$

$$\begin{array}{r} 0001 \quad 1 \\ 0010 \quad -2 \end{array}$$

\rightarrow

$$\begin{array}{r} 0001 \\ \underline{+1110} \\ 1111 \end{array}$$

$$1111$$

$$V=0$$

$$N=1$$

$$S=1$$

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

$$\neg \overline{V}$$

$$V=0$$

$$N=1$$

$$S=1$$