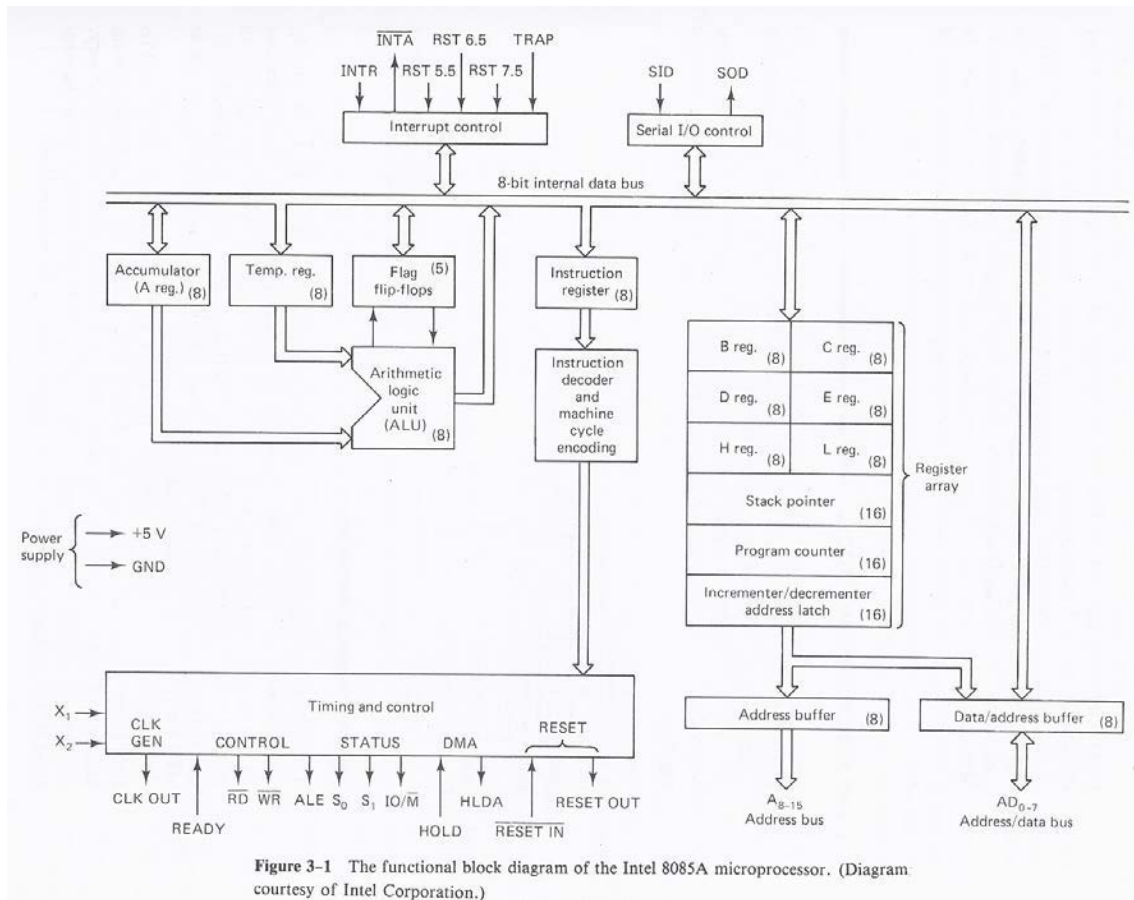


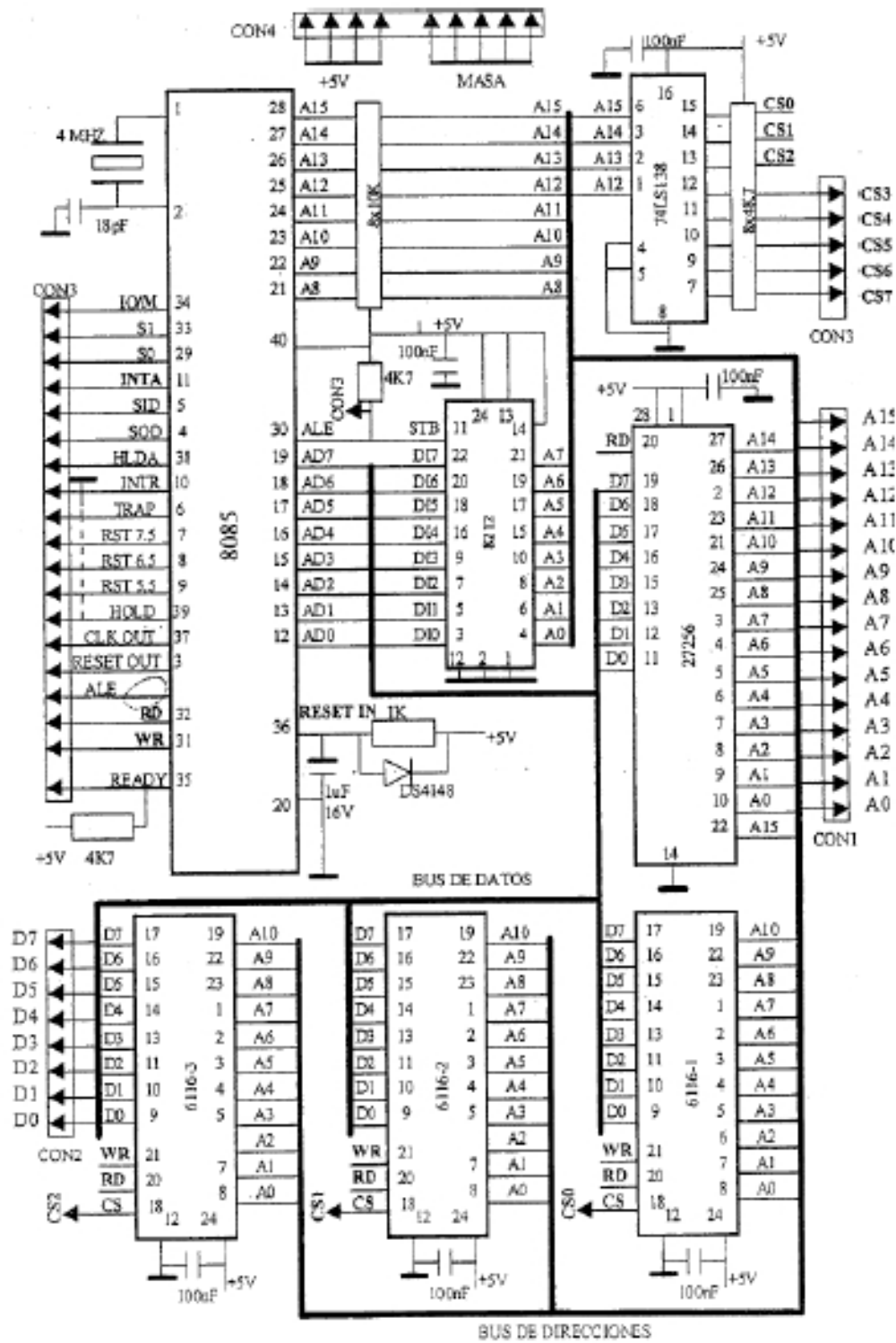
LABORATEGIKO PRAKTIKAK

8085

8085aren Kontrol Unitatea eta erregistroak



8085aren bloke diagrama



8085aren agindu joko

NEMONICO	EXP. GRAF.	FLAGS
INSTRUCCIONES DE TRANSFERENCIA		
MOV r1,r2	$(r1) \leftarrow (r2)$	NINGUNO
MOV r,M	$(r) \leftarrow [(HL)]$	NINGUNO
MOV M,r	$[(HL)] \leftarrow (r)$	NINGUNO
MVI r,byte	$(r) \leftarrow \text{byte}$	NINGUNO
MVI M,byte	$[(HL)] \leftarrow \text{byte}$	NINGUNO
LXI rp,doble	$(rpl) \leftarrow 1^\circ \text{ byte}$ $(rph) \leftarrow 2^\circ \text{ byte}$	NINGUNO
LDA addr	$(A) \leftarrow [addr]$	NINGUNO
STA addr	$[addr] \leftarrow (A)$	NINGUNO
LHLD addr	$(L) \leftarrow [addr]$ $(H) \leftarrow [addr+1]$	NINGUNO
SHLD addr	$[addr] \leftarrow (L)$ $[addr+1] \leftarrow (H)$	NINGUNO
LDAX rp	$(A) \leftarrow [(rp)]$	NINGUNO
STAX rp	$[(rp)] \leftarrow (A)$	NINGUNO
XCHG	$(H) \leftrightarrow (D)$ $(L) \leftrightarrow (E)$	NINGUNO
INSTRUCCIONES ARITMÉTICAS		
ADD r	$(A) \leftarrow (A) + (r)$	TODOS
ADD M	$(A) \leftarrow (A) + [(HL)]$	TODOS
ADI byte	$(A) \leftarrow (A) + \text{byte}$	TODOS
ADC r	$(A) \leftarrow (A) + (r) + CY$	TODOS
ADC M	$(A) \leftarrow (A) + [(HL)] + CY$	TODOS
ACI byte	$(A) \leftarrow (A) + \text{byte} + CY$	TODOS
SUB r	$(A) \leftarrow (A) - (r)$	TODOS
SUB M	$(A) \leftarrow (A) - [(HL)]$	TODOS
SUI byte	$(A) \leftarrow (A) - \text{byte}$	TODOS
SBB r	$(A) \leftarrow (A) - (r) - CY$	TODOS
SBB M	$(A) \leftarrow (A) - [(HL)] - CY$	TODOS
SBI byte	$(A) \leftarrow (A) - \text{byte} - CY$	TODOS
INR r	$(r) \leftarrow (r) + 1$	Z, S, P, AC
INR M	$[(HL)] \leftarrow [(HL)] + 1$	Z, S, P, AC
DCR r	$(r) \leftarrow (r) - 1$	Z, S, P, AC
DCR M	$[(HL)] \leftarrow [(HL)] - 1$	Z, S, P, AC
INX rp	$(rp) \leftarrow (rp) + 1$	NINGUNO
DCX rp	$(rp) \leftarrow (rp) - 1$	NINGUNO
DAD rp	$(HL) \leftarrow (HL) + (rp)$	CY
DAA	Ajuste BCD de (A)	NINGUNO
INSTRUCCIONES LÓGICAS		
ANA r	$(A) \leftarrow (A) \text{ and } (r)$ $(CY) \leftarrow 0, (AC) \leftarrow 1$	TODOS
ANA M	$(A) \leftarrow (A) \text{ and } [(HL)]$ $(CY) \leftarrow 0, (AC) \leftarrow 1$	TODOS
ANI byte	$(A) \leftarrow (A) \text{ and byte}$ $(CY) \leftarrow 0, (AC) \leftarrow 1$	TODOS
XRA r	$(A) \leftarrow (A) \text{ xor } (r)$	TODOS
XRA M	$(A) \leftarrow (A) \text{ xor } [(HL)]$	TODOS
XRI byte	$(A) \leftarrow (A) \text{ xor byte}$	TODOS
ORA r	$(A) \leftarrow (A) \text{ or } (r)$	TODOS
ORA M	$(A) \leftarrow (A) \text{ or } [(HL)]$	TODOS
ORI byte	$(A) \leftarrow (A) \text{ or byte}$	TODOS
CMP r	$(A) - (r)$	TODOS
CMP M	$(A) - [(HL)]$	TODOS
CPI byte	$(A) - \text{byte}$	TODOS

NEMONICO	EXP. GRAF.	FLAGS
INSTRUCCIONES DE ROTACIÓN Y FLAGS		
RLC	Rotación izqda	CY
RRC	Rotación dcha	CY
RAL	Rot. izqda. con CY	CY
RAR	Rot. dcha. con CY	CY
CMA	Comp. A1 de (A)	NINGUNO
CMC	Invierte (CY)	CY
STC	$(CY) \leftarrow 1$	CY
INSTRUCCIONES DE BIFURCACIÓN		
ccc=NZ salto si no cero (Z=0), ccc=Z salto si cero (Z=1), ccc=NC salto si no acarreo (CY=0), ccc=C salto si acarreo (CY=1), ccc=PO salto si paridad impar (P=0), ccc=PE salto si paridad par (P=1), ccc=P salto si positivo (S=0), ccc=M salto si negativo (S=1)		
JMP addr	$(PC) \leftarrow \text{addr}$	NINGUNO
Jccc addr	Si ccc=1, $(PC) \leftarrow \text{addr}$; Si ccc=0, $(PC) \leftarrow (PC) + 3$	NINGUNO
CALL addr	Guarda PC en la pila $(PC) \leftarrow \text{addr}$	NINGUNO
Cccc addr	Si ccc=1, guarda PC en la pila, $(PC) \leftarrow \text{addr}$; Si ccc=0, $(PC) \leftarrow (PC) + 3$	NINGUNO
RET	Recupera PC de la pila	NINGUNO
Rccc	Si ccc=1, recupera PC de la pila; Si ccc=0 $(PC) \leftarrow (PC) + 1$	NINGUNO
RSTn	$(PC) \leftarrow n \times 8$	NINGUNO
PCHL	$(PC) \leftarrow (HL)$	NINGUNO
INSTRUCCIONES DE MANEJO DE LA PILA		
PUSH rp	$[(SP)-1] \leftarrow (rpl)$ $[(SP)-2] \leftarrow (rph)$ $(SP) \leftarrow (SP) - 2$	NINGUNO
PUSH PSW	$[(SP)-1] \leftarrow (A)$ $[(SP)-2] \leftarrow (RE)$ $(SP) \leftarrow (SP) - 2$	NINGUNO
POP rp	$(rph) \leftarrow [(SP)]$ $(rpl) \leftarrow [(SP)+1]$ $(SP) \leftarrow (SP) + 2$	NINGUNO
POP PSW	$(RE) \leftarrow [(SP)]$ $(A) \leftarrow [(SP)+1]$ $(SP) \leftarrow (SP) + 2$	NINGUNO
XLTH	$(L) \leftrightarrow [(SP)]$ $(H) \leftrightarrow [(SP)+1]$	NINGUNO
SPLH	$(HL) \leftrightarrow (SP)$	NINGUNO
INSTRUCCIONES DE ENTRADA Y SALIDA		
IN puerta	$(A) \leftarrow [\text{puerta}]$	NINGUNO
OUT puerta	$[\text{puerta}] \leftarrow (A)$	NINGUNO
INSTRUCC. DE CONTROL DE INTERRUPCIONES		
EI	Habilita interrupciones	NINGUNO
DI	Inhabilita interrupciones	NINGUNO
HLT	Para el microprocesador	NINGUNO
NOP	No hace nada	NINGUNO
RIM	Lee línea serie y estado interrupciones.	NINGUNO
SIM	Escribe en línea serie y Programa interrupciones.	NINGUNO

RE=Status Register

PSW=Processor Status Word

- Hurrengo saioetan 8085aren aginduak landuko ditugu. Fijatzen joango gara, aginduetan ez ezik, baita ere hauek erregistro blokea nola eraldatzen duten, nola kargatzen den memorian programa baten kode bitarra, nola eraldatzen den programaren zenbatzailearen (Program Counter, PC) balioa,...
- Gogoratu bai memoria helbideak baita bertan dauden datuak ere 0 eta 1en sekuentzia luzeak direla, baina hamaseitarrean adieraziko direla ulergarritasuna errazteko.
- 8085aren bloke diagraman ikus daitekeenez, A erregistroa (metatzailea, 8 bitekoa) erregistro berezi bat da: edozein eragiketaren emaitza bertan biltegitratuko da (aurretik zeukan balioa galduz beraz). Hau da, eragiketa baten emaitza interesatzen bazaigu, eragiketa berri bat exekutatu aurretik nonbaiten gorde beharko dugu.
- F erregistroa ere berezia da: 8085aren 5 egoera bitak biltegitratzen ditu (jauzi eta adarkatze aginduetara iristean itzuliko gara hona).
- Orokorrean erregistroek byte bat biltegitratu dezakete, DE eta HL erregistroek binaka lan egin dezakete, 16 biteko erregistro bat balira (honela memoria helbide bat gorde dezakete).
- Beren mnemonikoan i bat duten aginduek, datua zehaztea behar duten aginduak dira (immediate). Ikusi adibidez MVI eta MOV-ren arteko aldea.
- Beren mnemonikoan x bat duten aginduek, erregistro bikoitzeko aginduak dira. Adibidez: INR vs INX. INR D aginduak D erregistroaren edukia gehituko du; INX D aginduak berriz, DE bikotea sortutako “16 biteko erregistroa” gehituko du (hau da, orokorrean E gehituko du, ez bada hau 00ra pasatzen bururakoaren eraginez D gehituz).
- Agindu baten atzetik datorren M hizkia ez da aldagai bat, mnemonikoaren parte da. M honek HL erakusleak adierazitako memoria posizioari egiten dio erreferentzia. Hau da $M=[HL]$.
- *Gelan erabiliko dugun simuladorean, makina kodea 8000h helbidetik aurrera kargatzen da; hortaz, guk 9000h helbidetik aurrera lan egingo dugu, programaren gainean ez idazteko.*

S4 Konputagailuen Egitura Laborategiak

8085 mihiztadura lengoaia: mugimendua, aritmetikoak, logikoak eta errotazioak (ptr.gabe)

Egin hurrengo ariketak. Bakoitzean kopia fitxategi batean programa mihiztadura lengoia eta aztertu mihizatzean sortzen den kode bitarra. Lehen bi ariketetarako sortu irudikoa bezalako taula bat eta exekutatu programak pausuz-pausu. Arreta jarri erregistroek (bereziki F eta PCak) hartzen dituzten balioei eta identifikatu agindu bakoitza bere makina kodearekin (ikusi nola agindu guztiek ez duten luzera bera).

Mnemonikoa	Memoria helbidea	Agindua (hex)

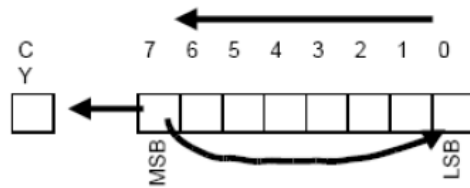
Kargatu egelan dagoen imagina 1.

- a) Gorde 9005h helbidean 07h.
- b) 9000h posizioan 4 elementuko array bat hasten da. Kopiatu ezazu array hori 9100h posiziotik aurrera alderantzizko ordenean.
- c) Murritzatu unitate batean 9100h posizioan (aurreko atalean sortutako) arrayaren elementu bakoitzaren balioa.
- d) Idatzi 9010h helbidean 9000htik aurrera dagoen arraya osatzen duten 4 elementuen baturaren emaitza.
- e) Gehitu (unitate batean) 9005h posizioan edukia.
- f) Batu 9001h eta 9101h posizioetako edukiak. Emaizta hamaseitarrean dago, pasa ezazu BCDra.

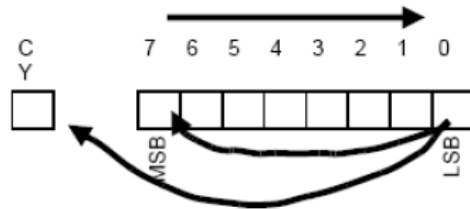
Kargatu egelan dagoen imajina 2.

- a) 910Ah helbidean BCD zenbaki baten pisurik txikieneko nibble-a dago; 910Bh-n berriz, esanguratsuenak. Elkartu BCD zenbakiaren bi digituak eta utzi emaitza 900Ah-n.
- b) Memoriako 9000h posizioan datu bat dago; gorde 900Bh eta 900Ch posizioetan hurrenez-hurren, zenbaki honen baterako eta birako osagarriak.
- c) Aurreko ataleko emaitzako imajina aldatu gabe, burutu hurrengo eragiketak: $[900Ah] - [9000h] \rightarrow [9101]$; $[900Ah] + [900Ch] \rightarrow [9102]$. Azkenik 9101h eta 9102h posizioetan dauden datuen XOR.
- d) 9000h posizioan 6 elementuko array bat hasten da. Positiboak edo negatiboak diren jakiteko, aplikatu ezazu maskara bat eta sortu array berri bat 9100h helbidetik aurrera, zeinak 00h balioa izango duen 910j posizioan 900j-ko zenbakia positiboa bazen; negatiboa izatekoan berriz, 80h.
- e) Konprobatu ea 900A-en dagoen zenbakia 31h baino handiagoa edo berdina den. Nola egiten da konparaketa? Non gordetzen da emaitza?
- f) Biderkatu birekin 11h zenbaki hamaseitarra errota agindu bat erabiliz eta gorde emaitza metagailuan. Emaizta aztertu zenbakia bitarrean idatziz.
- g) Exekutatu 1) eta 2) programak. Zer gertatzen da kasu bakoitzean?
 - a. `mvi A 81; ral; ral`
 - b. `mvi A 81; ral; rlc`

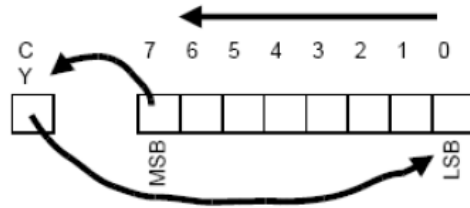
RLC



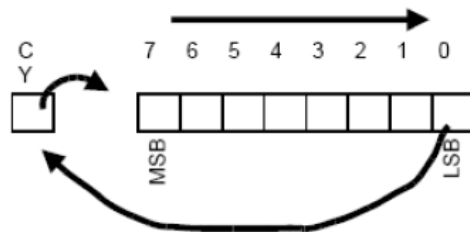
RRC



RAL



RAR



S5 Konputagailuen Egitura Laborategiak

8085 mihizadura lengoaia: + jauzi eta adarkatzeak

F S Z - AC - P - CY
 □ □ □ □ □ □ □ 2

Egoera ikurrak burutu berri den eragiketaren emaitzaren inguruko informazioa dute (gogoratu mugimendu aginduek ez dutela eraginik flag hauengan). Informazio hau exekuzioaren fluxua bideratzeko erabili daiteke (jauziak baldintzatu eta begiztak sortzeko).

Jauzi aginduak daude: hauen mnemonikoa J-z hasten da eta baldintzatuak izan daitezke ala baldintzarik gabekoak (JMP). Baldintzatutakoak izatekotan, J eta gero baldintza dator (z: zero; nz: ez zero, e.a.). Bota begirada bat. Mnemonikoa eta gero jauzia nora egin behar den zehaztu behar da (etiketa baten bitartez adieraziko dugu, mihiztatzaileak dagokion memoria helbidera bihurtuko duena). Jauzi agindu baten ondoren, programaren exekuzioak helbide berritik aurrera jarraitzen du.

Beste aukera bat azpierrutinari deiak dira (CALL). Jauzien antzekoak dira, baina azpierrutinaren exekuzioa amaitzean itzulera agindu batek (RET) programaren exekuzioa azpierrutina deitua izan zen puntura itzultzen du. Jauziekin ikusitakoaren antzera, azpierrutinari (-tik) deiak (itzulera) ere baldintzatuak izan daitezke ala ez.

Kargatu egelan dagoen imagina 3.

- a) Irakurri 9000h posizioko datua: zero ez bada, kopia ezazu 900Ah-n; zero bada idatzi FF.
- b) Irakurri 9001h eta 9002h-eko datuak: [9001]<[9002] bada, utzi dauden moduan, bestela trukatu hauen posizioak.
- c) 9000h posizioan 6 elementuko array bat hasten da, sortu ezazu 6 elementuko array berri bat 9100h-tik aurrera, zeinak aurreko arrayaren elementuei dagozkien paritate bikoiti bitak dituen (bytearen lsb posizioan).
- d) Irakurri 9000h-ko edukia eta 07h baina txikiagoa bada, gorde bere balioa 900Bh-n.
- e) Irakurri 9001h posizioko edukia eta 0Ah baina handiagoa edo berdina izatekoatan, gorde 0Fh 900Ch posizioan.

S5 Konputagailuen Egitura Laborategiak

8085 mihiztadura lengoia: erakusleak

Gorago aipatzen genuen 8085aren erregistroen byte bat biltegiratu dezaketela, baina batzuk binaka lan egin dezaketela (HL, DE), 16 biteko erregistro bat bezala. Honela, memoriaren tarte batean lan egin nahi bada, array bat dagoelako adibidea, kodea berrerabiltzeko aukera izan dezakegu. Ikus dezagun nola:

1. *“erregistro bikoitza” horretan hasierako helbidea kargatu (orain, erregistro hori memoria posizio horretara erakusle bat dela esango dugu)*
2. *nahi den eragiketa burutu*
3. *erregistroaren edukia gehitu*
4. *2. Pausura itzuli*

Modu honetan, eragiketa bera burutu behar bada arrayaren elementu guztiekin, ez dugu 2. pausu hori helbide berri bakoitzarentzat idatzi behar. Bestetik, beharrezkoa izango da ea amaitu dugun konprobatzea (betirako begiztan geratu nahi ez badugu).

1. *“erregistro bikoitza” horretan hasierako helbidea kargatu*
2. *nahi den eragiketa burutu*
3. *erregistroaren edukia gehitu*
4. *amaierara iritsi al gara? BAI 5; EZ 6*
5. *2. pausura itzuli*
6. *amaiera*

S4ko zenbait ariketa errepikatuko ditugu, baina orain erakusleak erabiliz (konparatu sortutako kodeak eta emaitzak).

Kargatu egelan dagoen imagina 1.

- a) 9000h posizioan 4 elementuko array bat hasten da. Murriztu unitate arrayaren elementu bakoitzaren balioa.
- b) Idatzi 9010h helbidean 9000htik aurrera dagoen arraya osatzen duten 4 elementuen baturaren emaitza.
- c) Gehitu (unitate batean) 9000h posizioan hasten den arrayaren 4 elementuetako bakoitza.
- d) . 9000h posizioan 4 elementuko array bat hasten da. Kopiatu ezazu array hori 9100h posiziotik aurrera alderantzizko ordenean.

S6 Konputagailuen Egitura Laborategiak

8085 mihiztadura lengoaia: erakusleak

Idatzi hurrengo eragiketak burutzeko programak. Aztertu kasu bakoitzean kodea mihizatzean sortzen den kode bitarra.

- e) Sortu ezazu 10 elementudun array bat hurrengo balioekin: 0, 1, 2, ..., 9. Arraya 9000h-9009h tartean egongo da.

Kargatu egelan dagoen imagina 2.

- f) 9000h posizioan 6 elementuko array bat hasten da. Positiboak edo negatiboak diren jakiteko, aplika ezazu maskara bat eta sortu array berri bat 9100h helbidetik aurrera, zeinak 00h balioa izango duen 910j posizioan 900j-ko zenbakia positiboa bazen; negatiboa izatekoan berriz, 80h.

S6 Konputagailuen Egitura Laborategiak

8085 mihiztadura lengoaia: programa osoak

Idatzi eta konprobatu hurrengo programak.

- a) Idatzi 9001h posizioan balioa etengabe irakurtzen duen programa eta, balioa 03h bada, eragiketarik gabeko bi atzerapen sartzen dituen ondoren, 3. atakan 01h
- b) Sistema batek 2 sentzore konprobatzen ditu eta hauen egoera adierazten duen bita (0 edo 1) kopiatzen du B erregistroko pisurik txikieneko 2 bitetan. Idatzi, beste 6 bitak kontutan hartu gabe, bi hauetako bat '1'ean dagoen konprobatzen duen programa. Baiezko kasuan, programak arreta azpierrutinara ("arreta") salto egingo du. Bertan 01 balio 1 atakan edo/eta 2 atakan jarriko da, gaituta egon den sentzorearen arabera. Ondoren B erregistroa ezabatuko du konprobaketara itzultzeko
- c) Memoriako 9000h posizioan array baten lehen elementua dago. Elementu guztiak balio positiboak direla jakinik, bilatu arraya osatzen duten 10 elementuetatik txikiena eta gorde bere posizio erlatiboa (arrayaren barruan duena) arrayaren azken elementuaren hurrengo memoria-posizioan..
- d) Gelako praktiketan, 2 zenbaki biderkatzeko eta baten faktoriala kalkulatzeko fluxu-diagramak egin ziren (GP3). Idatzi eta konprobatu 8085ean faktoriala kalkulatzeko kodea.
- e) Memoriako 9000h posizioan array baten lehen elementua dago. Horren luzera 10 elementukoa dela jakinik, idatzi arrayaren elementuak txikienetik handienera ordenatuko dituen programa.