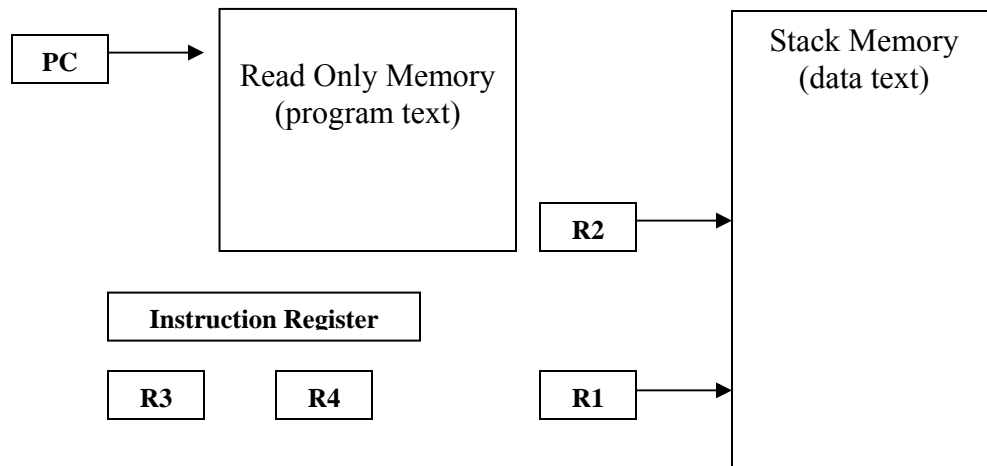


Pelajarilah arsitektur dari simulator Object Machine ST4469 berikut !



Secara garis besar, mesin objek ST4469 terdiri dari 3 (tiga) bagian :

1. Read Only Memory
Berfungsi menyimpan program yang akan dieksekusi
2. Stack Memory
Berfungsi untuk menyimpan data yang sifatnya Read and Write.
3. Kumpulan Register
Berfungsi sebagai pointer area didalam memory, baik pada ROM maupun Stack Memory. Register PC selalu menunjuk alamat pada ROM, yang fungsinya menunjuk instruksi selanjutnya yang akan dieksekusi. Register R2 berfungsi sebagai penunjuk stack top pada stack memory. Register R1 sebagai penunjuk base dari frame-stack pada stack memory (hal lebih rinci akan dibahas pada pemanggilan fungsi / procedure). Sedangkan register R3 dan R4 bersifat umum.

Instruksi Object Machine

Format

OPCODE *b i a*

b : base register *i* : indeks register, *a* : address

OPCODE sendiri terdiri dari 13 instruksi, yaitu :

lit, lod, sto, mvx, opr, int, cal, rtn, jmp, jpc, get, put, hlt

untuk penjelasan lebih rinci mengenai fungsi dan perilaku dari masing-masing instruksi akan dijelaskan nanti.

Object code selalu tersimpan pada area read only dan instruksi-instruksi selalu disimpan mulai dari alamat (address) 0 dan 1 intruksi lengkap (opcode maupun seluruh operand) tersimpan pada satu alamat. Object Machine mempunyai 6 register yaitu :

Register	Berfungsi Sebagai
PC	Program Counter
R1	Stack Base Pointer
R2	Stack Top Pointer
R3	Register Indeks
R4	Register Indeks
IR	Register Instruksi

Dengan catatan, register IR digunakan sebagai internal register oleh mesin objek, sehingga kita tidak pernah dapat mengubah isi / nilai-nya. Selain dari register IR, kita dapat mengubah secara langsung maupun tak langsung melalui instruksi mesin objek. Sedangkan nilai default dari masing-masing register (selain dari IR) ketika mesin mulai dijalankan adalah sebagai berikut :

PC = 0
 R1 = 0
 R2 = -1
 R3 = 0
 R4 = 0

Notasi $s[R2]$ menunjukkan isi dari stack dengan indeks dari R2.

Contoh :

- apabila R2 berisikan nilai 5, maka $s[R2]$ berarti $s[5]$, yaitu bagian dari stack yang beralamat 5.
- Notasi $s[R2] := 3$ berarti nilai 3 dimasukan ke bagian stack yang alamatnya ditunjuk oleh isi R2.

Berikut adalah perilaku rinci dari masing-masing instruksi pada mesin ST4469.

1. Load literal

lit $b\ i\ a$

$R2 := R2 + 1; s[R2] := 'a'$

Operand b, i diabaikan

2. Load

lod $b\ i\ a$

$R2 := R2 + 1; s[R2] := s[Rb + Ri + a]$

Isi dari register yang ditunjuk oleh b dan i dijumlahkan dengan a menghasilkan address untuk operand. Catatan, ketika $b = 0$ dan $i = 0$, register Rb dan Ri diabaikan.

3. Store

sto *b i a*

$s[Rb + Ri + a] := s[R2]; R2 := R2 - 1$

Isi dari register yang ditunjuk oleh *b* dan *i* dijumlahkan dengan *a* menghasilkan address untuk operand. Catatan, ketika *b* = 0 dan *i* = 0, register *Rb* dan *Ri* diabaikan.

4. Move to index register

mvx *b i a*

$Ri := s[R2]; R2 := R2 - 1$

Operand *b* diabaikan.

5. Arithmetic operation

opr *b i a*

<i>a</i>	
0	$s[R2] := -s[R2]$
1	$R2 := R2 - 1; s[R2] := s[R2] + s[R2 + 1]$
2	$R2 := R2 - 1; s[R2] := s[R2] - s[R2 + 1]$
3	$R2 := R2 - 1; s[R2] := s[R2] * s[R2 + 1]$
4	$R2 := R2 - 1; s[R2] := s[R2] \div s[R2 + 1]$
5	$R2 := R2 - 1; \text{Jika } s[R2] = s[R2 + 1] \text{ maka } s[R2] := 1, \text{ selain itu } s[R2] := 0$
6	$R2 := R2 - 1; \text{Jika } s[R2] \neq s[R2 + 1] \text{ maka } s[R2] := 1, \text{ selain itu } s[R2] := 0$
7	$R2 := R2 - 1; \text{Jika } s[R2] < s[R2 + 1] \text{ maka } s[R2] := 1, \text{ selain itu } s[R2] := 0$
8	$R2 := R2 - 1; \text{Jika } s[R2] \leq s[R2 + 1] \text{ maka } s[R2] := 1, \text{ selain itu } s[R2] := 0$
9	$R2 := R2 - 1; \text{Jika } s[R2] > s[R2 + 1] \text{ maka } s[R2] := 1, \text{ selain itu } s[R2] := 0$
10	$R2 := R2 - 1; \text{Jika } s[R2] \geq s[R2 + 1] \text{ maka } s[R2] := 1, \text{ selain itu } s[R2] := 0$

Operand *b* dan *i* diabaikan.

6. Increment stack top

int *b i a*

$R2 := R2 + a$

Operand *b* dan *i* diabaikan.

7. Procedure call

cal *b i a*

$s[R2 + 1] := R1; s[R2 + 2] := PC + 1;$

$s[R2 + 3] := R3; s[R2 + 2] := R4;$

$R1 := R3 + 5; R2 := R2 + 4; PC := a$

Operand *b* dan *i* diabaikan.

8. Return

rtn *b i a*

R2:=R1-5; PC:=s[R1-3];
R3:=s[R1-2]; R4:=s[R1-1]; R1:=s[R1-4]

Operand *b* dan *i* diabaikan.

9. Unconditional Jump

jmp *b i a*

PC:=*a*;

Operand *b* dan *i* diabaikan.

10. Jump on condition

jpc *b i a*

(jika s[R2] = 0 maka PC:=*a*;) R2 := R2-1

Operand *b* dan *i* diabaikan.

11. Read data

get *b i a*

R2 := R2+1 kemudian membaca 1 data bertipe integer dari keyboard, dan menyimpan data tersebut di s[R2]

Operand *b* dan *i* diabaikan.

12. Write data

put *b i a*

menampilkan s[R2] ke display dalam format desimal, kemudian R2 := R2-1.

Operand *b* dan *i* diabaikan.

13. Halt

hlt *b i a*

Operand *b*, *i* dan *a* diabaikan.

Tugas :

Dengan menggunakan instruksi dari mesin objek ST4469 :

1. Buatlah program yang menampilkan bilangan yang diinputkan oleh user !
2. Buatlah program yang mengambil inputan dari user dan menampilkan bilangan 5 jika input dari user ≤ 5 , dan menampilkan bilangan 10 apabila input dari user > 5 !
3. Buatlah program yang menampilkan bilangan dari 1 hingga 100 !
4. Buatlah program yang menampilkan bilangan dari 1 hingga N, dimana nilai N diambil dari user input !
5. Buatlah program menjumlahkan dua bilangan yang kedua bilangan tersebut diambil dari user input, dan tampilkan hasil penjumlahannya !
6. Buatlah program penjumlahan dari 1 hingga N, dimana Nilai N diambil dari user input dan hasil penjumlahan ditampilkan !
7. Buatlah program faktorial dari N, dimana nilai N diambil dari user input dan hasil perhitungan ditampilkan !