

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computación

Bachillerato en Ingeniería en Computación



IC-3101 Arquitectura de Computadoras

Profesor: Ing. Eduardo A. Canessa Montero, M.Sc.

Tarea Corta 6

Generación de microcódigo para una microarquitectura de procesador escalar

Estudiante:

Ericka Céspedes Moya 2017239557

II Semestre, 2018

Parte 1

Registros de propósito general

Para **add [R1:R2], AF, R7** y se usaron R1, R2 y R7 porque el problema lo requería. Además, R3 y R4 se utilizaron como registros temporales para almacenar la parte alta y la parte baja de la dirección 00AF.

jmp #ABC5 no utilizó registros de propósito general. Se cargaron los datos de la dirección directamente en el IP.

Para **or [AF03], [R1:R2], R6** se utilizaron los registros R1, R2 y R6 porque el problema lo requería. Al igual que el primero, se usaron los registros R3 y R4 como registros temporales para almacenar la parte alta y la parte baja de la dirección AF03.

Registros de propósito específico

Todos los ejercicios ocuparon del registro acumulador, el RA, el RB, el registro de banderas, el MAR y el MBR, excepto para el **jmp #ABC5** que no usó el RB porque no se requería. En el **jmp #ABC5** sólo se requirió el RA para hacer el incremento del MAR.

Parte 2

add [R1:R2], AF, R7

add [R1:R2], AF, R7

```
1) LD_ALL = 0, EN_ALL = 0
INC MAR (11) 2-12
2) LH_X = 0, EN_X = 1
3) LD_RA = 1, C = 2
4) LD_ACC = 1, EN_X = 0, LH_MAR = 0, IPD = 0, EN_ACC = 1, LD_RFLAGS = 1
5) LD_MAR = 1, LD_RA = 0, LD_ACC = 0
6) Si CFLAG = 0 //Salta a 12
7) LH_X = 1, EN_X = 1, EN_ACC = 0
8) LD_RA = 1, C = 2
9) LD_ACC = 1, EN_ACC = 1, EN_X = 0
10) LH_MAR = 1
11) LD_MAR = 1
12) LD_ALL = 0, EN_ALL = 0
13) MRW = 1, MDIR = 1, EN_MBR = 1
14) LD_MBR = 1
15) LD_R3 = 1 //R3 = Parte alta [00]
16) LD_MBR = 0, EN_MBR = 0, LD_R3 = 0
17) INC MAR(11) 17-27
28) MRW = 1, MDIR = 1, EN_MBR = 1
29) LD_MBR = 1
30) LD_R4 = 1 //R4 = Parte baja [AF]
31) LD_MBR = 0, EN_MBR = 0, LD_R4 = 0
// [00AF]
32) EN_R4 = 1, IPD = 0, LH_MAR = 0 //Parte baja R4
33) LD_MAR = 1 // MAR = [__AF]
34) EN_R4 = 0, EN_R3 = 1, LD_MAR = 0, LH_MAR = 1 //Parte alta R3
35) LD_MAR = 1 // MAR = [00AF]
36) EN_R3 = 0, MRW = 1, EN_MBR = 1, MDIR = 1 //MBR = [00AF]
37) LD_MBR = 1 //Bus de datos = [00AF]
38) LD_MBR = 0, LD_RA = 1 //RA = [00AF]

// RB = R7 y AF add R7
39) LD_ALL = 0, EN_ALL = 0
40) EN_R7 = 1
41) LD_RB = 1 // RB = R7
42) LD_RB = 0, EN_R7 = 0 // RB = R7
43) C = 0 // RA add RB
44) LD_ACC = 1, LD_RFLAGS = 1
45) LD_ACC = 0, LD_RFLAGS = 0
```

```

//Escribir en [R1:R2] el resultado de AF add R7
46) EN_R2 = 1 //Parte baja R2
47) IPD = 0, LH_MAR = 0
48) LD_MAR = 1 // MAR = [__:R2]
49) EN_R1 = 1, LD_MAR = 0 //Parte alta R1
50) LH_MAR = 1
51) LD_MAR = 1 // MAR = [R1:R2]
52) MRW = 0, MDIR = 0, EN_R1 = 0, EN_ACC = 1
53) LD_MBR = 1
54) MRW = 1 //Dato estable y escrito
55) Fetch

```

jmp #ABC5

```

jmp #ABC5

1) LD_ALL = 0, EN_ALL = 0
INC MAR (11) 2-12
2) LH_X = 0, EN_X = 1
3) LD_RA = 1, C = 2
4) LD_ACC = 1, EN_X = 0, LH_MAR = 0, IPD = 0, EN_ACC = 1, LD_RFLAGS = 1
5) LD_MAR = 1, LD_RA = 0, LD_ACC = 0
6) Si CFLAG = 0 //Salta a 12
7) LH_X = 1, EN_X = 1, EN_ACC = 0
8) LD_RA = 1, C = 2
9) LD_ACC = 1, EN_ACC = 1, EN_X = 0
10) LH_MAR = 1
11) LD_MAR = 1
12) LD_ALL = 0, EN_ALL = 0
13) MRW = 1, MDIR = 1, EN_MBR = 1
14) LD_MBR = 1
15) LH_IP = 1 //IP = Parte alta [1A]
16) LD_IP = 1
17) LD_MBR = 0, EN_MBR = 0, LD_IP = 0
18) INC MAR(11) 18-28
29) MRW = 1, MDIR = 1, EN_MBR = 1
30) LD_MBR = 1
31) LH_IP = 0 //IP = Parte baja [0B]
32) LD_IP = 1
33) LD_MBR = 0, EN_MBR = 0, LD_IP = 0
34) Fetch

```

or [AF03], [R1:R2], R6

or [AF03], [R1:R2], R6

```
1) LD_ALL = 0, EN_ALL = 0
INC MAR (11) 2-12
2) LH_X = 0, EN_X = 1
3) LD_RA = 1, C = 2
4) LD_ACC = 1, EN_X = 0, LH_MAR = 0, IPD = 0, EN_ACC = 1, LD_RFLAGS = 1
5) LD_MAR = 1, LD_RA = 0, LD_ACC = 0
6) Si CFLAG = 0 //Salta a 12
7) LH_X = 1, EN_X = 1, EN_ACC = 0
8) LD_RA = 1, C = 2
9) LD_ACC = 1, EN_ACC = 1, EN_X = 0
10) LH_MAR = 1
11) LD_MAR = 1
12) LD_ALL = 0, EN_ALL = 0
13) MRW = 1, MDIR = 1, EN_MBR = 1
14) LD_MBR = 1
15) LD_R3 = 1 //R3 = Parte alta [AF]
16) LD_R3 = 0, LD_MBR = 0, EN_MBR = 0
17) INC MAR(11) 17-27
28) MRW = 1, MDIR = 1, EN_MBR = 1
29) LD_MBR = 1
30) LD_R4 = 1 //R4 = Parte baja [03]
31) LD_R4 = 0, LD_MBR = 0, EN_MBR = 0

// RA = [R1:R2] y RB = R6
32) EN_R6 = 1
33) LD_RB = 1 // RB = R6
34) LD_RB = 0, EN_R6 = 0 // RB = R6
// [R1:R2]
35) EN_R2 = 1, IPD = 0, LH_MAR = 0 //Parte baja R2
36) LD_MAR = 1 // MAR = [__:R2]
37) EN_R2 = 0, EN_R1 = 1, LD_MAR = 0, LH_MAR = 1 //Parte alta R1
38) LD_MAR = 1 // MAR = [R1:R2]
39) EN_R1 = 0, MRW = 1, EN_MBR = 1, MDIR = 1 //MBR = [R1:R2]
40) LD_MBR = 1 //Bus de datos = [R1:R2]
41) LD_RA = 1, C = 7
42) LD_MBR = 0, EN_MBR = 0, LD_ACC = 1, LD_RFLAGS = 1
43) LD_ACC = 0, LD_RFLAGS = 0
```

```
//Escribir en [AF03] el resultado de [R1:R2] OR R6 que está en ACC
44) EN_R4 = 1 //R4 = Parte baja [03]
45) IPD = 0, LH_MAR = 0
46) LD_MAR = 1 // MAR = [__03]
47) EN_R3 = 1, LD_MAR = 0 //R3 = Parte alta [AF]
48) LH_MAR = 1
49) LD_MAR = 1 // MAR = [AF03]
50) MRW = 0, MDIR = 0, EN_R3 = 0, EN_ACC = 1
51) LD_MBR = 1
52) MRW = 1 //Dato estable y escrito
53) Fetch
```

Parte 3

add [R1:R2], AF, R7

add [R1:R2], AF, R7

```
1) LD_ALL = 0, EN_ALL = 0
2) INC_MAR = 1
3) INC_MAR = 0, MRW = 1, MDIR = 1, EN_MBR = 1
4) LD_MBR = 1
5) LD_R3 = 1 //R3 = Parte alta [00]
6) LD_MBR = 0, EN_MBR = 0, LD_R3 = 0
7) INC_MAR = 1
8) INC_MAR = 0, MRW = 1, MDIR = 1, EN_MBR = 1
9) LD_MBR = 1
10) LD_R4 = 1 //R4 = Parte baja [AF]
11) LD_MBR = 0, EN_MBR = 0, LD_R4 = 0
// [00AF]
12) EN_R4 = 1, IPD = 0, LH_MAR = 0 //Parte baja R4
13) LD_MAR = 1 // MAR = [__AF]
14) EN_R4 = 0, EN_R3 = 1, LD_MAR = 0, LH_MAR = 1 //Parte alta R3
15) LD_MAR = 1 // MAR = [00AF]
16) EN_R3 = 0, MRW = 1, EN_MBR = 1, MDIR = 1 //MBR = [00AF]
17) LD_MBR = 1 //Bus de datos = [00AF]
18) LD_MBR = 0, LD_RA = 1 //RA = [00AF]

// RB = R7 y AF add R7
19) LD_ALL = 0, EN_ALL = 0
20) EN_R7 = 1
21) LD_RB = 1 // RB = R7
22) LD_RB = 0, EN_R7 = 0 // RB = R7
23) C = 0 // RA add RB
24) LD_ACC = 1, LD_RFLAGS = 1
25) LD_ACC = 0, LD_RFLAGS = 0

//Escribir en [R1:R2] el resultado de AF add R7
26) EN_R2 = 1 //Parte baja R2
27) IPD = 0, LH_MAR = 0
28) LD_MAR = 1 // MAR = [__:R2]
29) EN_R1 = 1, LD_MAR = 0 //Parte alta R1
30) LH_MAR = 1
31) LD_MAR = 1 // MAR = [R1:R2]
32) MRW = 0, MDIR = 0, EN_R1 = 0, EN_ACC = 1
33) LD_MBR = 1
34) MRW = 1 //Dato estable y escrito
35) Fetch
```

jmp #ABC5

jmp #ABC5

- 1) LD_ALL = 0, EN_ALL = 0
- 2) INC_MAR = 1
- 3) INC_MAR = 0, MRW = 1, MDIR = 1, EN_MBR = 1
- 4) LD_MBR = 1
- 5) LH_IP = 1 //IP = Parte alta [1A]
- 6) LD_IP = 1
- 7) LD_MBR = 0, EN_MBR = 0, LD_IP = 0
- 8) INC_MAR = 1
- 9) INC_MAR = 0, MRW = 1, MDIR = 1, EN_MBR = 1
- 10) LD_MBR = 1
- 11) LH_IP = 0 //IP = Parte baja [0B]
- 12) LD_IP = 1
- 13) LD_MBR = 0, EN_MBR = 0, LD_IP = 0
- 14) Fetch

or [AF03], [R1:R2], R6

or [AF03], [R1:R2], R6

```
1) LD_ALL = 0, EN_ALL = 0
2) INC_MAR = 1
3) INC_MAR = 0, MRW = 1, MDIR = 1, EN_MBR = 1
4) LD_MBR = 1
5) LD_R3 = 1 //R3 = Parte alta [AF]
6) LD_R3 = 0, LD_MBR = 0, EN_MBR = 0
7) INC_MAR = 1
8) INC_MAR = 0, MRW = 1, MDIR = 1, EN_MBR = 1
9) LD_MBR = 1
10) LD_R4 = 1 //R4 = Parte baja [03]
11) LD_R4 = 0, LD_MBR = 0, EN_MBR = 0

// RA = [R1:R2] y RB = R6
12) EN_R6 = 1
13) LD_RB = 1 // RB = R6
14) LD_RB = 0, EN_R6 = 0 // RB = R6
// [R1:R2]
15) EN_R2 = 1, IPD = 0, LH_MAR = 0 //Parte baja R2
16) LD_MAR = 1 // MAR = [__:R2]
17) EN_R2 = 0, EN_R1 = 1, LD_MAR = 0, LH_MAR = 1 //Parte alta R1
18) LD_MAR = 1 // MAR = [R1:R2]
19) EN_R1 = 0, MRW = 1, EN_MBR = 1, MDIR = 1 //MBR = [R1:R2]
20) LD_MBR = 1 //Bus de datos = [R1:R2]
21) LD_RA = 1, C = 7
22) LD_MBR = 0, EN_MBR = 0, LD_ACC = 1, LD_RFLAGS = 1
23) LD_ACC = 0, LD_RFLAGS = 0

//Escribir en [AF03] el resultado de [R1:R2] OR R6 que está en ACC
24) EN_R4 = 1 //R4 = Parte baja [03]
25) IPD = 0, LH_MAR = 0
26) LD_MAR = 1 // MAR = [__03]
27) EN_R3 = 1, LD_MAR = 0 //R3 = Parte alta [AF]
28) LH_MAR = 1
29) LD_MAR = 1 // MAR = [AF03]
30) MRW = 0, MDIR = 0, EN_R3 = 0, EN_ACC = 1
31) LD_MBR = 1
32) MRW = 1 //Dato estable y escrito
33) Fetch
```

Parte 4

add [R1:R2], AF, R7

add [R1:R2], AF, R7

- 1) LD_ALL = 0, EN_ALL = 0
- 2) INC_MAR = 1
- 3) INC_MAR = 0, MRW = 1, MDIR = 1, EN_MBR = 1
- 4) LD_MBR = 1
- 5) LD_RA = 1 //RA = [00AF]
- 6) LD_MBR = 0, EN_MBR = 0, LD_RA = 0

// RB = R7 y AF add R7

- 20) EN_R7 = 1
- 21) LD_RB = 1 // RB = R7
- 22) LD_RB = 0, EN_R7 = 0 // RB = R7
- 23) C = 0 // RA add RB
- 24) LD_ACC = 1, LD_RFLAGS = 1
- 25) LD_ACC = 0, LD_RFLAGS = 0

//Escribir en [R1:R2] el resultado de AF add R7

- 26) EN_R2 = 1 //Parte baja R2
- 27) IPD = 0, LH_MAR = 0
- 28) LD_MAR = 1 // MAR = [__:R2]
- 29) EN_R1 = 1, LD_MAR = 0 //Parte alta R1
- 30) LH_MAR = 1
- 31) LD_MAR = 1 // MAR = [R1:R2]
- 32) MRW = 0, MDIR = 0, EN_R1 = 0, EN_ACC = 1
- 33) LD_MBR = 1
- 34) MRW = 1 //Dato estable y escrito
- 35) Fetch

jmp #ABC5

jmp #ABC5

- 1) LD_ALL = 0, EN_ALL = 0
- 2) INC_MAR = 1
- 3) INC_MAR = 0, MRW = 1, MDIR = 1, EN_MBR = 1
- 4) LD_MBR = 1
- 5) LD_IP = 1 // IP = [1A0B]
- 4) LD_MBR = 0, EN_MBR = 0, LD_IP = 0
- 5) Fetch

or [AF03], [R1:R2], R6

or [AF03], [R1:R2], R6

```
1) LD_ALL = 0, EN_ALL = 0
2) INC_MAR = 1
3) INC_MAR = 0, MRW = 1, MDIR = 1, EN_MBR = 1
4) LD_MBR = 1
5) LD_R3 = 1 //R3 = [AF03]
6) LD_R3 = 0, LD_MBR = 0, EN_MBR = 0

// RA = [R1:R2] y RB = R6
7) EN_R6 = 1
8) LD_RB = 1 // RB = R6
9) LD_RB = 0, EN_R6 = 0 // RB = R6
// [R1:R2]
10) EN_R2 = 1, IPD = 0, LH_MAR = 0 //Parte baja R2
11) LD_MAR = 1 // MAR = [__:R2]
12) EN_R2 = 0, EN_R1 = 1, LD_MAR = 0, LH_MAR = 1 //Parte alta R1
13) LD_MAR = 1 // MAR = [R1:R2]
14) EN_R1 = 0, MRW = 1, EN_MBR = 1, MDIR = 1 //MBR = [R1:R2]
15) LD_MBR = 1 //Bus de datos = [R1:R2]
16) LD_RA = 1, C = 7
17) LD_MBR = 0, EN_MBR = 0, LD_ACC = 1, LD_RFLAGS = 1
18) LD_ACC = 0, LD_RFLAGS = 0

//Escribir en [AF03] el resultado de [R1:R2] OR R6 que está en ACC
19) EN_R3 = 1 //R3 = [AF03]
20) IPD = 0
21) LD_MAR = 1 // MAR = [AF03]
22) MRW = 0, MDIR = 0, EN_R3 = 0, EN_ACC = 1
23) LD_MBR = 1
24) MRW = 1 //Dato estable y escrito
25) Fetch
```