

Examen d'architecture des ordinateurs

Solution

1-A Il faut commencer par lire l'opérande 2 en mémoire et le mettre dans un registre, car les opérations arithmétiques ne peuvent être réalisées qu'entre 2 registres : $r12 \leftarrow [rad2]$

1-B l'opération précédente sera réalisée à l'entrée d'un état supplémentaire que l'on nommera par exemple `load_op2`. Le tableau suivant est une représentation de la partie du graphe demandée :

Transition	Decode \rightarrow load_op2	Load_op2 \rightarrow pcplus1
condition	/ir[31]*ir[15]	1
action	$R12 \leftarrow [rad2]$	$Rdest \leftarrow Rs1 \text{ op } R12$

1-C

	Areg	Breg	Dreg	Ualcmd	Dbusin	write
Decode \rightarrow load_op2	lr[19..16]	0000	1100	0000	10	0
Load_op2 \rightarrow pcplus1	lr[23..20]	1100	lr[27..24]	lr[31..28]	01	0

2- Nombre d'occurrences d'un entier dans un tableau

PILE = 0x100

M = 10

```
    set  PILE, %sp
    set  tab, %r1
    set  M, %r2
    set  2, %r3
    call nb_occurrences
stop: ba  stop
tab:   .word 3, 2, 7, 5, 2, 11, 6, 9, 4, 2
```

// nombre d'occurrences d'un élément x dans un tableau

// IN : r1 = adresse du tableau

// IN : r2 = nombre d'éléments

// IN : r3 = valeur x

// OUT : r4 = nombre d'occurrences trouvées

nb_occurrences:

```
    push %r5
    push %r6
    clr  %r4
    clr  %r5  // index
loop: cmp  %r5, %r2
      bgeu end_loop
      ld   [%r1+%r5], %r6
      cmp  %r6, %r3
```

```

        bne suite
        inc %r4
suite:   inc %r5
        ba loop
end_loop:
        pop %r6
        pop %r5
        ret

```

3- Affichage dynamique de la valeur d'un compteur

```

NB_CYCLES = 6
PILE = 0x100
LEDS = 0xB0000000

```

```

        ba progp
handler: push %r20
        // compeur dans r19
        set LEDS,%r20
        st %r19, [%r20]
        pop %r20
        reti

progp:  set PILE,%sp
        clr %r19
boucle: cmp %r19, NB_CYCLES-1
        bne incrementer
        clr %r19
        ba boucle
incrementer: inc %r19
        ba boucle

```