

Exo 2:

étiqu	Instructions	Commentaire
	1) Reset, S/I = 0 2) $C[31] \leftarrow 1$ 3) $M[31] \leftarrow 1$, 0 ailleurs 4) $Cpt = 0$ 5) SET 6) Ecriture	initialisation à 1 de tous les bits de poids fort des mots
Ped-0	7) $C[Cpt] \leftarrow 0$, $C[31] \leftarrow 1$ 8) $M[Cpt] \leftarrow 1$, $M[31] \leftarrow 1$ 9) SET 10) Recherche 11) Si (S/N = 0) Aller à Ped-1 12) $C[Cpt] \leftarrow 1$ 13) $C[31] \leftarrow 0$ 14) Ecriture	Same for C[31] on a déjà mis M[31] à 1 et 0 ailleurs
Ped-1	15) $C[Cpt] \leftarrow 1$, $C[31] \leftarrow 1$ 16) SET 17) Recherche 18) Si (S/N = 0) Aller à Fin 19) $C[Cpt] \leftarrow 0$ 20) $C[31] \leftarrow 1$ 21) Ecriture 22) $M[Cpt] \leftarrow 0$ 23) $Cpt = Cpt + 1$ 24) Si (Cpt = 31) Aller à Fin 25) Aller à Ped-0	
Fin	26) STOP	

Fin

13) SET

14) Recherche

15) Lecture

16) STOP

On peut lire la lecture car si le max a un 0 en bit poids fort alors $S/N = 0$ c'est tout les bits du reg I sont à 0 \Rightarrow on refait donc une recherche

nb de cycles nécessaire :

On sait que : nb de cycles moyen =

$$\frac{\text{cas favorable} + \text{cas défavorable}}{2}$$

* cas fav = si le max = '01...1' c'est on fait 63 itérations

* cas defav : si le max = '000...0' on fait toujours 63 itérations

\Rightarrow Dans cette solutions, tous les cas ont la même complexité d'où :

$$\text{nb cycle M} = \frac{63 + 63}{2} = 63 \text{ cycles}$$

* pour un processeur de 2GHz \Rightarrow c'est 2×10^9 cycles/seconde

$$\text{Temp d'exécution} = \frac{63}{2 \times 10^9} \approx \boxed{30 \text{ ns}}$$

Exo 3: Algo qui recherche le max :

Algo 1:

étiqu	Instructions	Commentaire
Bcl	1) Reset 2) $Cpt = 62$ 3) $C \leftarrow 011...1$ 4) $M \leftarrow 0$ partout 5) $M[Cpt] \leftarrow 1$ 6) SET 7) Recherche 8) Si (S/N = 1) Aller à suite 9) $C[Cpt] \leftarrow 0$ 10) $Cpt = Cpt - 1$ 11) Si (Cpt = -1) Aller à Fin 12) Aller à Bcl	Like yezed yezid won fel registre magique that's why!!!
suite		

Algo 2:

étiqu	Instructions	Commentaire
Bcl	1) Reset 2) $ACC \leftarrow 2^{62}$ 3) $M \leftarrow 1$ partout 4) $C \leftarrow ACC$ 5) SET 6) Recherche 7) Si (S/N = 1) Aller à Fin 8) $ACC = ACC - 1$ 9) Aller à Bcl	si on veut affecter qch chose à C, on utilise tjrs l'accumulateur (Archit 6.5) (minim +
Fin	10) Lecture 11) STOP	

2

nb de cycles nécessaire :

$$\text{nb de cycle moyen} = \frac{1 + 2^{63}}{2} = 2^{62}$$

⇒ avec un processeur de 2GHz, on aura :

$$\text{Temps d'exécution} = \frac{1}{2^{31}} \approx 68 \text{ ans.}$$

Exo 4 :

1) Algo qui permet de lire tous les nb sup ou égaux à 16 : (La question 2 est la même que exo 1)

étq	Instructions	Commentaire
Bcl	1) Reset 2) $C[4] \leftarrow 1$ 3) $M[4] \leftarrow 1$, '0' ailleurs 4) $S/I \leftarrow 0$ 5) SET 6) Recherche 7) SI ($S/N=0$) Aller à Fin 8) $S/I \leftarrow 1$ 9) Lecture, $BT \leftarrow 1$ 10) Aller à Bcl 11) STOP	
Fin		

Exo 5 :

étq	Instructions	Commentaire
	1) Reset-1, Reset-2 2) $C_2 \leftarrow '00 \dots 0'$ 3) $M_2 \leftarrow '11 \dots 1'$ 4) SET-2 5) Ecriture-2	initialisation de la 2 ^{ème} mémoire A à '0'
pair	6) $C_1[0] \leftarrow '0'$ 7) $M_1[0] \leftarrow '1'$, '0' ailleurs 8) $S/I-1 \leftarrow 0$ 9) SET-1 10) Recherche-1 11) SI ($S/N_1=0$) Aller à impair 12) $S/I-1 \leftarrow 1$ 13) Lecture-1, $BT_1 \leftarrow 1$ 14) $C_2 \leftarrow '00 \dots 0'$ 15) $S/I-2 \leftarrow 0$ 16) SET-2 17) Recherche-2 18) $S/I-2 \leftarrow 1$	On recherche le 1 ^{er} mot libre dans le m2

19) $C_2 \leftarrow RS_1$

20) Ecriture-2, $BT_2 \leftarrow 1$

21) Aller à pair

impair

22) $C_1[0] \leftarrow '1'$

23) $S/I-1 \leftarrow 0$

24) SET-1

25) Recherche-1

26) SI ($S/N-1=0$) Aller à Fin

27) $C_2 \leftarrow '0 \dots 0'$

28) $S/I-2 \leftarrow 0$

29) SET-2

30) Recherche-2

31) $S/I-2 \leftarrow 1$

32) $C_2 \leftarrow RS_1$

33) Ecriture-2, $BT_2 \leftarrow 1$

34) Aller à impair

Fin.

35) STOP

Exo 6 :

1) le schéma détaillé :

La mémoire d'entrée : FIFO

Mémoire de services : Associative

Les équations des bascules de préférences pour la

$$\text{FIFO} = \begin{cases} J_i = \text{Load}_i = BP_{i-1} \cdot BP_i \\ K_i = BP_i \cdot \overline{BP_{i+1}} \end{cases}$$

pour l'étape 0 : $J_0 = DE \cdot \overline{BP_0}$

pour l'étape n : $K_n = DL \cdot BP_n$

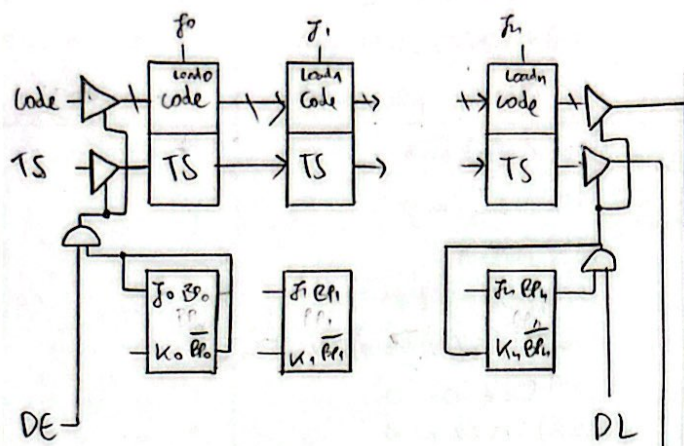
DE = Demande d'écriture :

DL = Demande de lecture

CE = Cond d'écriture = $DE \cdot \overline{BP_0} = J_0$

CL = Cond de lecture = $DL \cdot BP_n = K_n$

3



3) Algo incrémentation s'il existe déjà :

étq	instructions	commentaire
	1) Reset 2) $C.code \leftarrow code$ 3) $M.code \leftarrow '1'$, '0' ailleurs 4) SET 5) Recherche 6) lecture ($BT=0$) 7) $n = PS.nbr + 1$ 8) $C.nbr \leftarrow n$ 9) $M.nbr \leftarrow '1'$, '0' ailleurs 10) Ecriture ($BT=0$) 11) STOP	-il ne va y avoir qu'un seul répondant

→ $BT=0$ lorsqu'on veut accéder encore à la case! , $BT=1$ sinon.

2) Algo d'insertion d'un nv client qui n'existe pas dans la mémoire asso d'un service :

étq	instructions	commentaire
	1) Reset 2) $C \leftarrow '00 \dots 0'$ 3) $M \leftarrow '1'$ partout 4) $S/I \leftarrow 0$ 5) SET 6) Recherche 7) Si ($S/N=0$) Aller à Fin 8) $C.code \leftarrow code$ 9) $C.nbr \leftarrow 1$ 10) $S/I \leftarrow 1$ 11) Ecriture, $BT=1$ 12) $S/I \leftarrow 0$ 13) STOP	On suppose que la mémoire est à '0'
Fin		

4