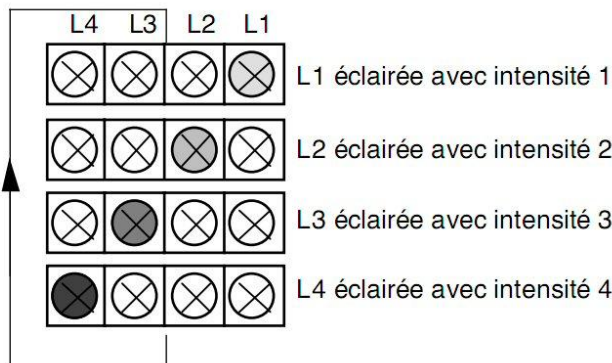


Travaux dirigés (Systèmes à processeurs n°2)

Synthèse

1. Etude d'un chenillard

Un chenillard est réalisé avec 4 lampes qui s'éclairent successivement avec une intensité comprise entre 1 et 4. Le cycle de fonctionnement est représenté par le schéma ci-dessous.



L'algorithme qui réalise cette animation est :

```

Faire pour toujours
  Début
    - Eclairer L1 avec intensité 1
    - Eclairer L2 avec intensité 2
    - Eclairer L3 avec intensité 3
    - Eclairer L4 avec intensité 4
  Fin
    
```

On dispose d'un sous-programme ALLUMER qui permet d'éclairer pendant 2 secondes une lampe avec une intensité fixée.



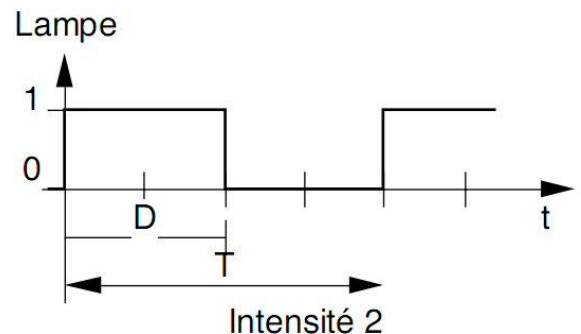
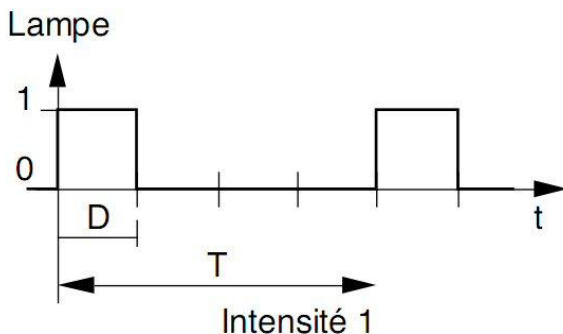
Remarque :

- R0.B détruit
- R1.B détruit
- Autres registres inchangés

Coder l'algorithme en source assembleur (pas de programme objet)

2. Variation de l'intensité lumineuse

Pour faire varier l'éclairage d'une lampe, on allume et on éteint la lampe avec un rapport cyclique variable de 1/4 à 1. La période T du phénomène est faible devant la durée de la persistance rétinienne.



Rapport cyclique : $C = \frac{n \cdot D}{T}$ n intensité lumineuse

Pour éclairer ou éteindre une lampe pendant une durée donnée, on a besoin d'une temporisation.

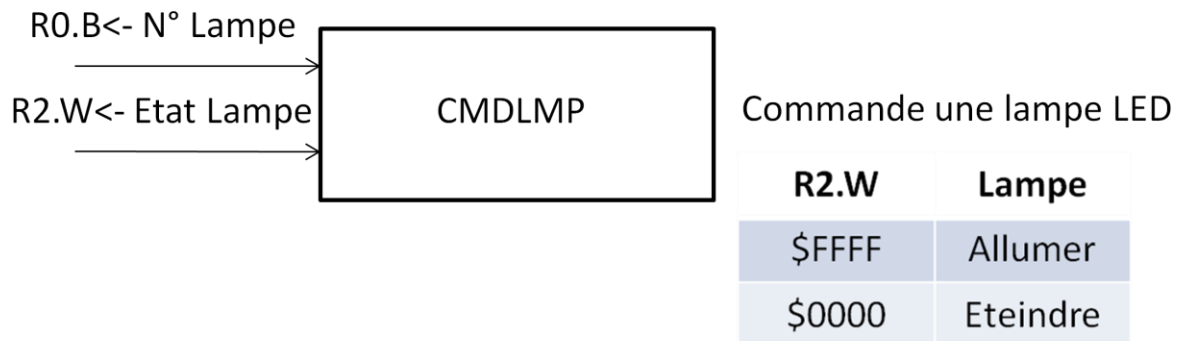
Une temporisation de durée D est réalisée par le sous programme suivant :

Etiquette	Opération	Opérande
TMP :	PUSHC	FLG
	MOV.W :G	R0,-2[SP]
JQCQ :	MOV.B :G	#7fh,R0I
	SUB.B :G	#1,R0I
	JNE	JQCQ
	MOV.W:G	-2[SP],R0
	JNE	FLG
	RTS	

- 2.1. Ce sous programme détruit il des registres ? justifier votre réponse
- 2.2. Etablir le programme objet correspondant à la temporisation
- 2.3. Evaluer le temps d'exécution (fréquence d'horloge du microprocesseur :32MHz)

3. Commande d'un éclairage

Pour commander une lampe, on dispose du sous programme suivant :



Remarque : pas de registres détruits

L'emploi du sous programme en notation algorithmique est le suivant :

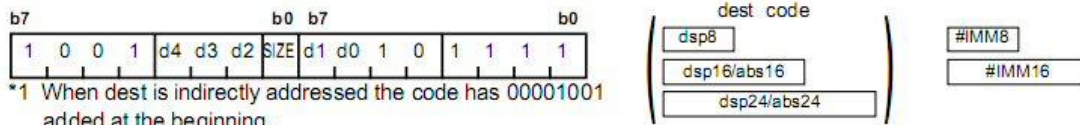
Début	Début
R0.B ← n	R0.B ← n
R2 ← \$FFFF	R2 ← 0
Exécuter CMDLMP	Exécuter CMDLMP
Fin	Fin

Etablir l'algorithme du programme qui permet d'éclairer la lampe n°2 avec une intensité de 3. Coder cet algorithme en assembleur.

On dispose des sous programmes :

- Temporiser pendant une durée D : TMP,
- Commander la lampe n°n :CMDLMP.

(1) MOV.size:G #IMM,dest



*1 When dest is indirectly addressed the code has 00001001 added at the beginning.

.size	SIZE	dest					dest												
		d4	d3	d2	d1	d0	d4	d3	d2	d1	d0								
.B	0	Rn					dsp:8[SB]												
.W	1						Rn					dsp:8[FB]							
An												dsp:16[A0]							
												dsp:16[A1]							
[An]					dsp:16[SB]														
					dsp:16[FB]														
dsp:8[An]					dsp:24[A0]														
					dsp:24[A1]														
dsp:8[An]					abs16														
					abs24														

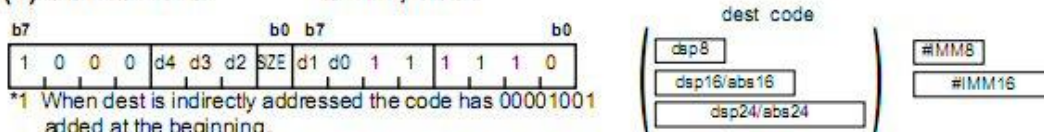
[Number of Bytes/Number of Cycles]

dest	Rn	An	[An]	dsp:8[An]	dsp:8[SB/FB]	dsp:16[An]	dsp:16[SB/FB]	dsp:24[An]	abs16	abs24
Bytes/Cycles	3/1	3/1	3/2	4/2	4/2	5/2	5/2	6/2	5/2	6/2

*2 When dest is indirectly addressed, the number of bytes and cycles in the table are increased by 1 and 3 respectively.

*3 When (.W) is specified for the size specifier(.size) the number of bytes in the table is increased by 1.

(1) SUB.size:G #IMM, dest



*1 When dest is indirectly addressed the code has 00001001 added at the beginning.

.size	SIZE	dest					dest												
		d4	d3	d2	d1	d0	d4	d3	d2	d1	d0								
.B	0	Rn					dsp:8[SB]												
.W	1						Rn					dsp:8[FB]							
An												dsp:16[A0]							
												dsp:16[A1]							
[An]					dsp:16[SB]														
					dsp:16[FB]														
dsp:8[An]					dsp:24[A0]														
					dsp:24[A1]														
dsp:8[An]					abs16														
					abs24														

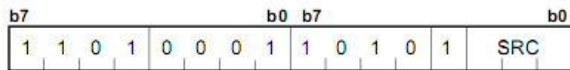
[Number of Bytes/Number of Cycles]

dest	Rn	An	[An]	dsp:8[An]	dsp:8[SB/FB]	dsp:16[An]	dsp:16[SB/FB]	dsp:24[An]	abs16	abs24
Bytes/Cycles	3/1	3/1	3/3	4/3	4/3	5/3	5/3	6/3	5/3	6/3

*2 When dest is indirectly addressed, the number of bytes and cycles in the table are increased by 1 and 3 respectively.

*3 When (.W) is specified for the size specifier(.size) the number of bytes in the table is increased by 1.

(1) PUSHC src

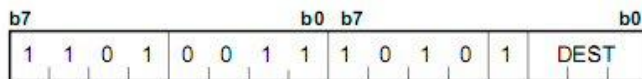


src	SRC	src	SRC
DCT0	0 0 0	DRC0	1 0 0
DCT1	0 0 1	DRC1	1 0 1
FLG	0 1 0	DMD0	1 1 0
SVF	0 1 1	DMD1	1 1 1

[Number of Bytes/Number of Cycles]

Bytes/Cycles	2/1
--------------	-----

(1) POPC dest



dest	DEST	dest	DEST
DCT0	0 0 0	DRC0	1 0 0
DCT1	0 0 1	DRC1	1 0 1
FLG	0 1 0	DMD0	1 1 0
SVF	0 1 1	DMD1	1 1 1

[Number of Bytes/Number of Cycles]

Bytes/Cycles	2/3
--------------	-----