

Examen : systèmes embarqués

J.-M Friedt, 2 janvier 2011

Tying the whole thing together was an embedded digital computer, made out of exotic devices called “integrated circuits” – silicon chips, running a set of esoteric programs. In the middle of the instrument panel, amid familiar dials and switches, stood the computer interface, a numeric keypad glowing with segmented digits. Throughout the mission the astronauts punched in numbers, ran programs, and read displays.

Digital Apollo – Human and Machine in Spaceflight, D.A. Mindell (MIT Press, 2008)

1 Conception d'un ordinateur : adressage d'une mémoire

Un ordinateur est, pour sa version la plus simple, un processeur entouré de quelques circuits de mise en œuvre (gestion d'horloge, de la réinitialisation), une mémoire non-volatile dans laquelle chercher les instructions du programme à exécuter, et une mémoire volatile dans laquelle conserver les données temporaires et variables.

Le Zilog Z80, développé en 1976, a été un processeur 8 bits longtemps utilisé dans les applications d'ordinateurs personnels (Radio Shack TRS80, Sinclair ZX81, Amstrad CPC, calculatrices Texas Instruments ...) et industrielles, avec aujourd'hui une version embarquée se démarquant notamment par l'ajout d'interfaces ethernet (eZ80).

Nous nous proposons d'étudier une mise en œuvre possible de ce processeur pour réaliser un petit ordinateur capable d'obtenir des instructions depuis une mémoire flash programmée au préalable, et de stocker en RAM le résultat de ses calculs. Pour ce faire, le processeur est entouré d'une mémoire flash 28C256 (256 kbits de mémoire), une RAM 6264 (64 kbits de capacité), et pour la gestion de périphériques deux latches 74HC574.

1.1 Rappel du brochage de quelques circuits

Pour rappel, quelques données des constructeurs de ces composants sont fournies ci-dessous :

- Table logique de fonctionnement de la mémoire flash **28C256** qui donne accès en lecture uniquement aux octets indexés par l'adresse A0..A14

Table 3. Operations ⁽¹⁾

	V _{FP}	Operation	\bar{E}	\bar{G}	\bar{W}	A9	DQ0 - DQ7
Read Only	V _{FPL}	Read	V _{IL}	V _{IL}	V _{IH}	A9	Data Output
		Output Disable	V _{IL}	V _{IH}	V _{IH}	X	Hi-Z
		Standby	V _{IH}	X	X	X	Hi-Z
		Electronic Signature	V _{IL}	V _{IL}	V _{IH}	V _{ID}	Codes
Read/Write ⁽²⁾	V _{FPH}	Read	V _{IL}	V _{IL}	V _{IH}	A9	Data Output
		Write	V _{IL}	V _{IH}	V _{IL} Pulse	A9	Data Input
		Output Disable	V _{IL}	V _{IH}	V _{IH}	X	Hi-Z
		Standby	V _{IH}	X	X	X	Hi-Z

Notes: 1. X = V_{IL} or V_{IH}
2. Refer also to the Command Table

DESCRIPTION
The M28F256 FLASH Memory is a non-volatile memory which may be erased electrically at the chip level and programmed byte-by-byte. It is organized as 32K bytes of 8 bits. It uses a command register architecture to select the operating modes and thus provides a simple microprocessor interface. The M28F256 FLASH Memory is suitable for applications where the memory has to be reprogrammed in the equipment. The access time of 100ns makes the device suitable for use in high speed microprocessor systems.

Table 1. Signal Names

A0..A14	Address Inputs
DQ0 - DQ7	Data Inputs / Outputs
\bar{E}	Chip Enable
\bar{G}	Output Enable
\bar{W}	Write Enable
V _{PP}	Program Supply
V _{CC}	Supply Voltage
V _{SS}	Ground

Figure 1. Logic Diagram

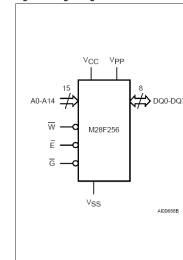
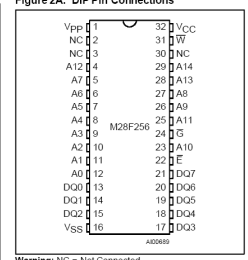


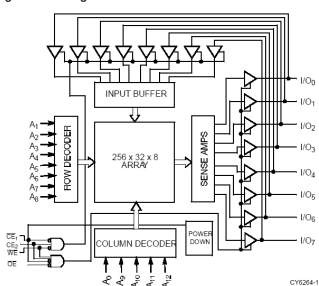
Figure 2A. DIP Pin Connections



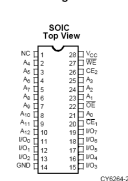
Warning: NC = Not Connected

- Table logique et brochage de la **RAM 6264** qui donne accès aux octets, en lecture ou écriture (signaux WE# et OE#), indexés par l'adresse A0..A12

Logic Block Diagram



Pin Configuration



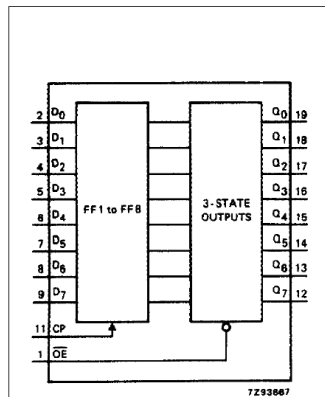
Truth Table

\bar{CE}_1	\bar{CE}_2	WE	\bar{OE}	Input/Output	Mode
H	X	X	X	High Z	Deselect/Power-Down
X	L	X	X	High Z	Deselect
L	H	H	L	Data Out	Read
L	H	L	X	Data In	Write
L	H	H	H	High Z	Deselect

- Table logique et brochage du **latch 74574** qui a pour rôle de mémoriser l'état en entrée D sur ses sorties Q lors d'un front d'horloge

Octal D-type flip-flop; positive edge-trigger;
3-state

74HC/HCT574



FUNCTION TABLE

OPERATING MODES	INPUTS			INTERNAL FLIP-FLOPS	OUTPUTS Q ₀ to Q ₇
	OE	CP	D _n		
load and read register	L	↑	l	L	L
register	L	↑	h	H	H
load register and disable outputs	H	↑	l	L	Z
	H	↑	h	H	Z

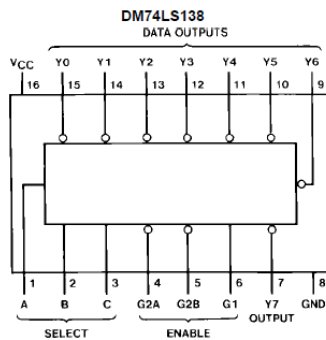
Notes

1. H = HIGH voltage level
h = HIGH voltage level one set-up time prior to the LOW-to-HIGH CP transition
L = LOW voltage level
l = LOW voltage level on set-up time prior to the LOW-to-HIGH CP transition
Z = HIGH impedance OFF-state
↑ = LOW-to-HIGH clock transition

- Table logique et brochage du **démultiplexeur 74138** qui a pour rôle d'activer par un signal actif au niveau bas une sortie Y d'indice égale à la valeur sur ses entrées A, B, C

Connection Diagrams

Function Tables



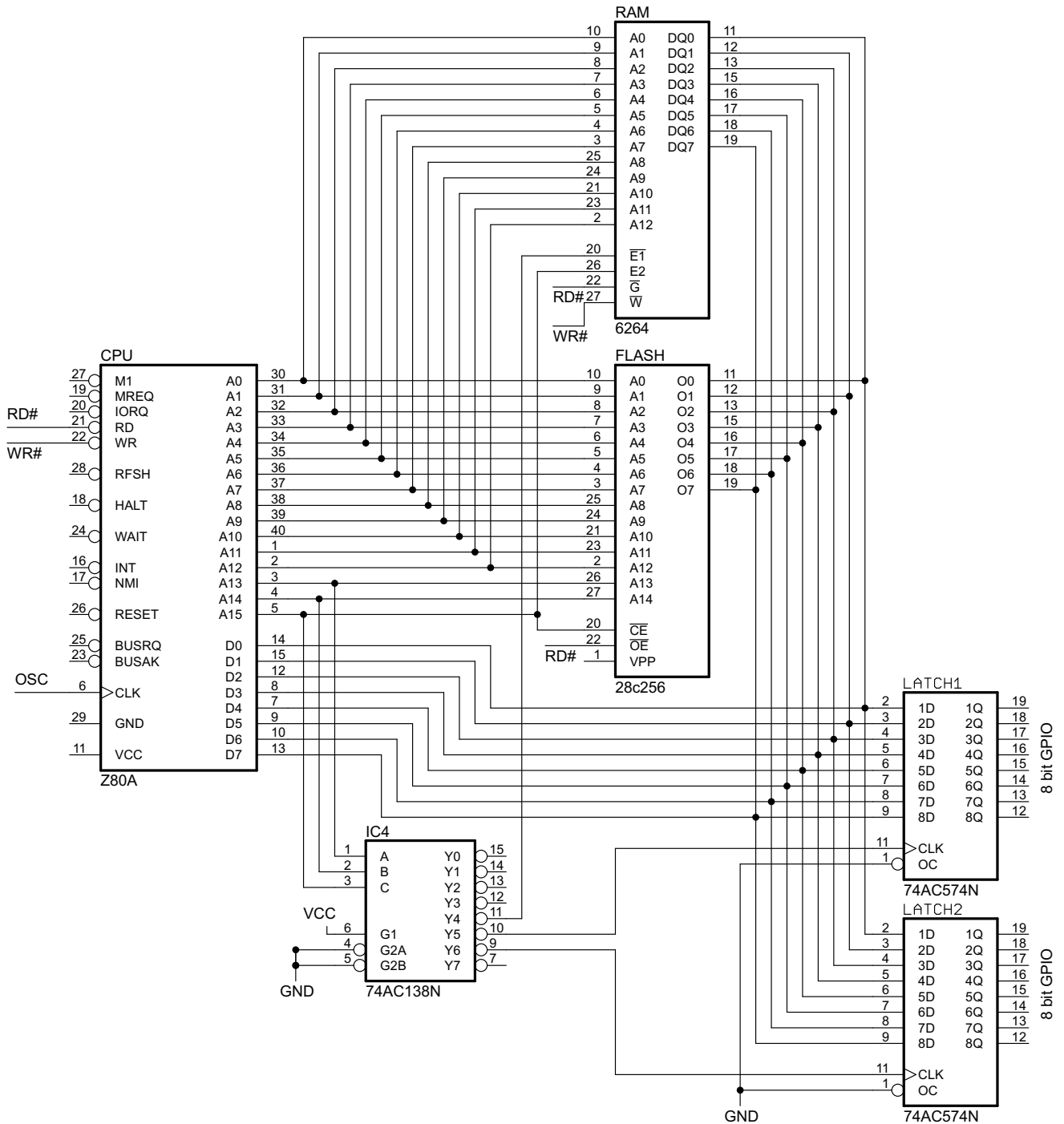
Inputs		Select		Outputs								
G1	G2 (Note 1)	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	L	H	H	H	H	H	H	H
H	L	L	H	L	H	L	H	H	H	H	H	H
H	L	L	H	H	L	H	L	H	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

Inputs		Select		Outputs			
Enable	G	B	A	Y0	Y1	Y2	Y3
H	X	X	X	H	H	H	H
L	X	X	X	L	H	H	H
L	L	L	L	L	L	H	H
L	L	L	H	L	L	L	H
L	L	H	L	L	L	L	L
L	L	H	H	L	L	L	L
L	H	L	L	L	L	L	L
L	H	L	H	L	L	L	L
L	H	H	L	L	L	L	L
L	H	H	H	L	L	L	L

H = HIGH Level
L = LOW Level
X = Don't Care
Note 1: G2 = G2A + G2B

1.2 Analyse du circuit

Ayant sélectionné les composants qui entoureront le processeur, nous implémentons le circuit suivant que nous allons analyser : il s'agit d'un circuit fonctionnel à l'exception du générateur d'horloge qui n'est pas explicité.



1. Quelle est la condition d'activation de la mémoire flash ? de la RAM ?
2. Quelle condition doit être vérifiée par le bus d'adresses du processeur (A0-A15) pour activer la flash ? pour activer la RAM ?
3. Quelle est l'espace adressable par ce processeur ? combien d'octets de mémoire sont ainsi accessibles ?
4. Compte tenu de la seconde réponse, sur quelle plage d'adresses le processeur accède-t-il à la mémoire flash ? à la RAM ?
5. Sachant que par conception, un Z80 s'initialise avec l'index de compteur (PC) à zéro, le processeur est-il capable d'aller chercher en mémoire non-volatile les premiers opcodes nécessaires à son initialisation ?
6. Rappeler la définition de GPIO : à quoi sert-il ? Le 74H574 est un latch : quelle est sa condition d'activation ?
7. Quelles plages d'adresses permettent d'activer les deux latches ?

2 Autonomie d'un circuit embarqué

Un microcontrôleur servant de séquenceur programmable pour la mise en marche périodique d'un instrument consomme continuellement 1 mA pour une tension d'alimentation de 3,3 V. L'instrument, alimenté sous 9 V, est mis en marche pendant 5 minutes toutes les deux heures : hors tension il ne consomme pas de courant, sous tension il consomme 500 mA.

L'ensemble des ces circuits est alimenté par une unique batterie de voiture (12 V) de capacité 22 A.h

1. Quelle est l'autonomie de ce circuit si les tensions d'alimentation du microcontrôleur et de l'instrument sont régulées par des régulateurs linéaires ?
2. Quelle est l'autonomie de ce circuit si les tensions d'alimentation du microcontrôleur et de l'instrument sont régulées par des régulateurs à découpage (DC-DC) ?

Nous ajoutons sur notre batterie un panneau solaire qui fournit en moyenne 60 mA pendant 8 h/jour.

1. Mêmes questions que plus haut.
2. Estimer la taille d'un tel panneau solaire.

3 Gestion de la pile

Un compilateur C exploite la pile pour stocker ses variables et passer les arguments aux fonctions.

```
1 #include <stdio.h>
  #include <stdlib.h>
3
4 int addition(int a, int b) {return(a+b);}
5
6 int main()
7 {char c[4];
  int i,k=2;
9 i=0;
11 printf("%d %d %d %d\n", sizeof(long long), sizeof(int), sizeof(short), sizeof(char));
  c[0]=0; c[1]=1; c[2]=2; c[3]=3; c[4]=4;
13 printf("i : %d\n", i);
15 k=addition(i, c[8]);
  printf("k : %d\n", k);
17
19 }
```

Listing 1 – Programme à analyser

L'exécution de ce programme sur un PC comprenant un processeur de type Intel x86, exécutant le système d'exploitation GNU/Linux, fournit le résultat suivant :

```
jmfriedt@eee:~/exam$ gcc -Wall -o exam exam.c
jmfriedt@eee:~/exam$ ./exam
8 4 2 1
i : 4
k : 6
```

Cependant sur un iPod – ordinateur basé sur un processeur de type ARM Cortex-A8 et exécutant le système d'exploitation iPhone OS X – le même programme ¹ fournit la sortie suivante :

```
iPod-touch:/Applications/HelloToolchain.app root# ./HelloToolchain
8 4 2 1
i : 0
k : 0
```

1. Expliquer la valeur de `i` lors de l'exécution du programme dans les deux cas.
2. Expliquer la valeur de `k` lors de l'exécution du programme dans les deux cas.
3. Expliquer la différence entre les deux architectures.

¹L'installation de la toolchain de crosscompilation basée sur GCC à destination des iPhone et iPod, exploitant les bibliothèques propriétaires fournies par Apple, est décrite à <http://code.google.com/p/iphonedevonlinux>

Solutions :

1 Conception d'un ordinateur : adressage d'une mémoire

16 bits d'adresse donc 65536 mots adressables. Le bus de données est sur 8 bits donc 65536 octets.

La RAM fait 64 kbits ou 8 koctets. La mémoire flash fait 256 kbits ou 32 koctets.

L'accès à la flash nécessite que $CE\#=0$, *i.e.* $A15=0$: la plage d'adresses est $0x0000-0x8000$.

L'accès à la RAM nécessite que $E2=1$ et $E1\#=0$, *i.e.* $A15=1$ et $Y4=0$. Or $Y4=0$ signifie que $A=B=0$ et $C=1$, *i.e.* $A13=0$ et $A14=0$. Donc la plage d'adresses de la RAM est $0x8000-0x9FFF$.

Les GPIOs s'accèdent par $Y5$ et $Y6$ passant au niveau bas respectivement, *i.e.* $C=1$ et $A=1/B=0$ ou $A=0/B=1$. Puisque A est connecté à $A13$ et B à $A14$, nous avons $A15=A14=1$ et $A13=0$ pour GPIO2, et $A15=A13=1$ et $A14=0$, soit $0xC000$ à $0xDFFF$ et $0xA000$ à $0xBFFF$. Ces deux plages d'adresses activent respectivement GPIO2 et GPIO1.

2 Autonomie d'un circuit embarqué

1 mA pendant 1 h et 0,5 A toutes les deux heures pendant 5 minutes qui est équivalent à 0,25 A pendant 5 minutes toutes les heures. Donc chaque heure, on consomme en moyenne $1\text{ mA} + 250 \times (5/60) = 22\text{ mA}$.

Lors de la conversion linéaire, ces 22 mA qui alimentent les circuits sont puisés directement sur la batterie : l'autonomie du circuit est donc de 1000 h, ou 41 jours.

Lors de la conversion DC-DC, nous travaillons à puissance constante, donc nous puisons

– pour le processeur, 3,3 mW soit 0.275 mA sous 12 V

– pour l'instrument, 4,5 W soit 0.375 A sous 12 V

Une fois les courants puisés sur la batterie calculés, l'autonomie se déduit de la même façon : chaque heure, le circuit puise en moyenne sur la batterie $0,275\text{ mA} + 375/2 \times (5/60) = 16\text{ mA}$. L'autonomie du circuit est donc cette fois 1375 heures ou 57 jours.

Avec le panneau solaire : nous ramenons en moyenne 20 mA à toute heure pour compenser les pertes de la batteries. Dans le cas de l'alimentation par convertisseur linéaire, la batterie doit donc fournir 2 mA et l'autonomie passe à 11000 heures ou 458 jours ou 1 an et 3 mois. Dans le cas de l'alimentation par convertisseur DC-DC, les pertes sont compensées par la charge et l'autonomie n'est limitée que par la détérioration de la batterie et des panneaux solaires.

Un panneau solaire capable de fournir 1 W sous 12 V occupe typiquement une surface de l'ordre de 100 à 300 cm².

3 Gestion de la pile

`char c[4]` ; alloue 4 octets, indexés de 0 à 3.

`c[4]` dépasse donc l'espace mémoire alloué, et la donnée est stockée dans l'espace mémoire adjacent, à savoir `i`.

De la même façon, `c[8]` accède à l'indice de départ du tableau `c+8`. Puisque chaque variable occupe 4 octets, il s'agit donc de l'emplacement de `k` qui vaut 2.