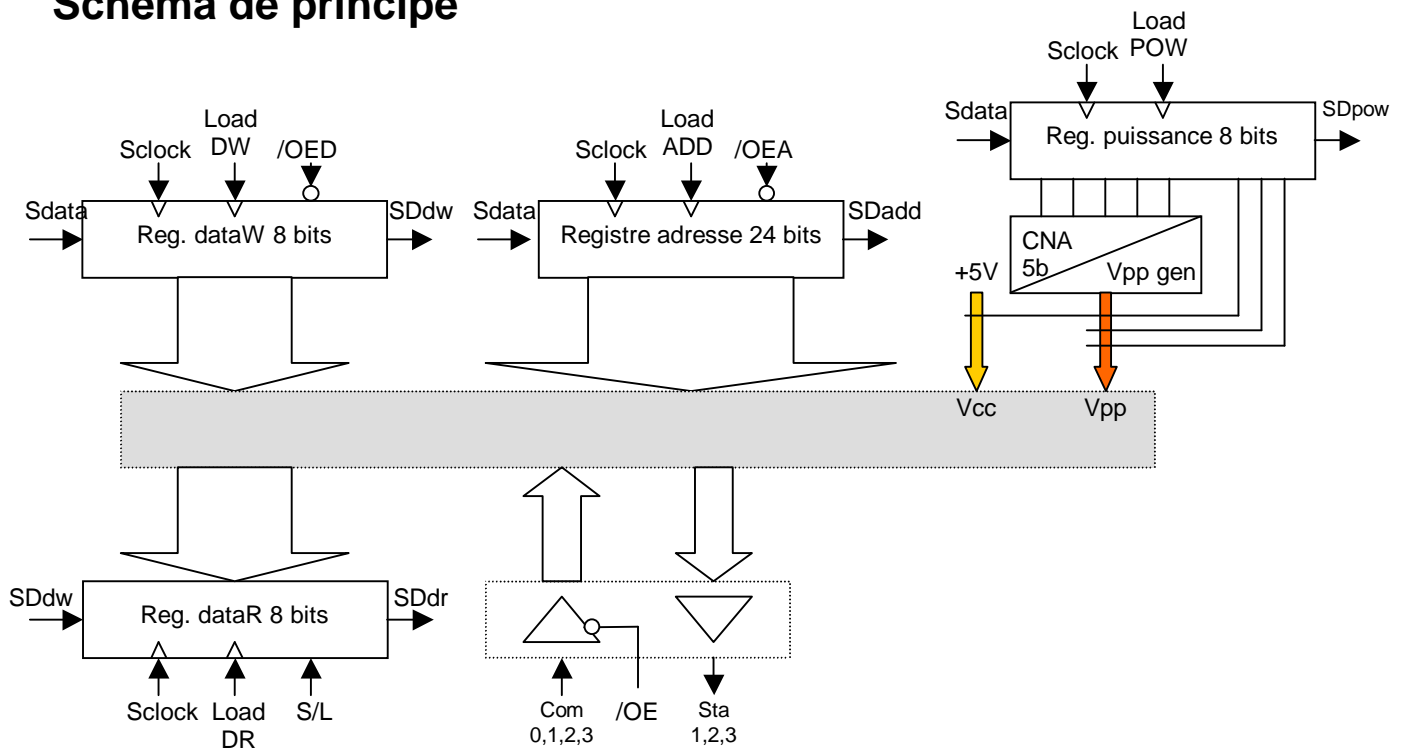


Schéma de principe



Tous les signaux de contrôles "rapides" sont accessibles en direct sur le bus de données ou sur les bits de status.

Les autres signaux sont relatifs aux registres de l'interface.

Signal	Mnémo	Sens	LPT	def	
Commande 0	Com0	In	D0	0	Buffer 74x241 contrôlé par OEC
Commande 1	Com1	In	D1	0	Buffer 74x241 contrôlé par OEC
Commande 2	Com2	In	D2	0	Buffer 74x241 contrôlé par OEC
Commande 3	Com3	In	D3	0	Buffer 74x241 contrôlé par OEC
Autorisation sortie dataW	OED	In	/D7	0	
Autorisation sortie adresses	OEA	In	/D6	0	
Autorisation sortie commandes	OEC	In	D4	0	Ces deux signaux pourraient être couplés
Status 1	Sta1	Out	+S6		Buffer 74x241
Status 2	Sta2	Out	+S5		Buffer 74x241
Status 3	Sta3	Out	+S4		Buffer 74x241
Horloge série	Sclock	In	D5	0	Front montant, signal raidi par un buffer 74x241
Données série	Sdata	In	D0	0	bit de poids fort sorti en premier
Chargement data W	LDW	In	-C1	0	Front montant
Chargement data R	LDR	In	-C0	0	Front montant
Shift/load data R	SL	In	+C0	1	=0 pour charger le registre, = LDR inversé
Load adresse	LADD	In	+C2	0	Front montant
Load puissance	LPOW	In	-C3	0	Front montant
Données série sortie dataW	SDdw	Out			Rebouclée sur entrée de DataR
Données série sortie adresse	SDadd	Out	-S7		
Données série sortie dataR	SDdr	Out	+S3		
Données série sortie puissance	SDpow	Out	???		Pas de connexion pour l'autotest

Architecture logicielle

PopManual	LoadAddress24 LoadAddress16 IncAddress LoadDataW LoadDataR	SetControl ClrControl PulseControl NPulseControl	WriteData WriteControl	_outp
CalibrationVpp	PowerOff PowerVccOn PowerVppZero PowerVppOn SetVpp	GetStatus PulseSclock ShiftOut ShiftIn	ReadData ReadControl ReadStatus	_inp
Interfacelnit	ShiftTest	ShiftInA24 ShiftInD16		
CalibrationTimer CalibrationSpeed				

Utilisation des contrôles

Les signaux de contrôle direct du circuit sont :

Com0, Com1, Com2, Com3 4 signaux en sorties, autorisés en sortie par EOC
 Sta1, Sta2, Sta3 3 signaux en entrées
 OED, OEA Autorisation des données en écriture et des adresses

Fonctions élémentaires :

Routine	iMask	Fonction	signaux utilisés
SetControl(iMask)	Com0, Com1, Com2, Com3, OEC, OEA, OED	Mise à 1 du signal logique iMask	D0 à D7
ClrControl(iMask)		Mise à 0 du signal logique iMask	D0 à D7
PulseControl(iMask)		Impulsion 010 du signal logique iMask	D0 à D7
NPulseControl(iMask)		Impulsion 101 du signal logique iMask	
GetStatus(iMask)	Sta1, Sta2, Sta3	Lecture du bit de status sta1 sta2 sta3	+S4 ,+S5,+S6

Utilisation des registres séries ; fonctions de base

Routine	Fonction	signaux utilisés
PulseSclock(iWord)	Envoie un front montant sur Sclock	Sclock
ShiftOut(iWord,nbits)	charger N bits dans les registres de sortie, MSB first	Sdata, Sclock
ShiftIn (iNbits)	acquérir les N bits du registre dataR+dataW, LSB first	SDdr, Sclock
ShiftInD16(void)	acquisition de 16 bits sur SDdr, MSB first (self test)	SDadd, Sclock
ShiftInA24(void)	acquisition de 24 bits sur SDadd, MSB first (self test)	SDadd, Sclock
ShiftTest (void)	Vérifier les boucles des registres dataR, dataW et adresses	Sdata, SDdr, SDadd, Sclock

Utilisation des registres séries ; fonctions évoluées

Routine	Fonction
---------	----------

LoadAddress24(int iAdd)	Chargement de iAdd dans le registre adresse Décalage des 24 bits, activation du load
LoadAddress16(int iAdd)	Chargement de iAdd dans le registre adresse Décalage des 16 bits, activation du load
IncAddress(void)	Incrément de l'adresse, chargement et load
LoadDataW(int iData)	Chargement de iData dans le registre dataW Décalage de 8 bits, activation du load
LoadDataR()	Chargement du registre DataR avec les données lues, puis tranfert série Activation du load puis décalage de 8 bits,

Utilisation des registres séries ; puissance

Routine	Fonction
PowerOff(void)	Vcc off
PowerVccOn(void)	Vcc=+5V, Vpp pulled up to Vcc
PowerVppZero(void)	Vcc=+5V, Vpp=0
PowerVppOn (void)	Vcc=+5V, Vpp=VPP
int GetPowerStatus(void)	retour état de l'alimentation
SetVpp (float fVpp)	Règle l'alim Vpp sur fVpp
float GetVpp(void)	retour de la valeur courante du Vpp

Fonctions diverses ; initialisation, calibrations

Routine	Fonction
bool Interfacelnit(hWnd,lptport,sefltest)	Initialisation de l'interface
bool CalibrationVpp()	calibration du générateur de Vpp
bool CalibrationTimer ()	calibration des fonctions timers
bool CalibrationSpeed()	calibration de la vitesse d'accès aux ports LPT
PopMANUAL()	activation du panneau de controle manuel

Format Intel

Un fichier « Intel hex » est un fichier imprimable compris d'un nombre quelconque de lignes au format Intel. Tous les champs sont en hexadécimal, avec 2 caractères ASCII par octet.

```

:2000000012014c75a800e4f508f509780a7a78e4f608dafcd283fcfded240af9a7050dbd81
:2000200000010ced2488ec34ff50edc283e4fcfded240af9e76d7013ed33e43c700d0dbd2a
:2000400000010ced2488ec34ff50e50509e50970020508e50924a8e50834fd50aee4f50874

:040000000001403289
:01000B0032C2
:03001300D21732CF
:01001B0032B2
:03002300020800D0
:03002B000212C2FC
:100040000074FFF580F590F5A0F5B075D0007587C8

:10130000D20A0BBBF0187B00D20B0CBC18107C006F
:101310000DBDE001E4FCCF2401CFCE3400CED0E0FF
:03132000D0D032F8
:00000001FF
    
```

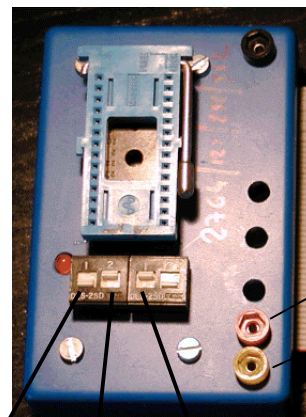
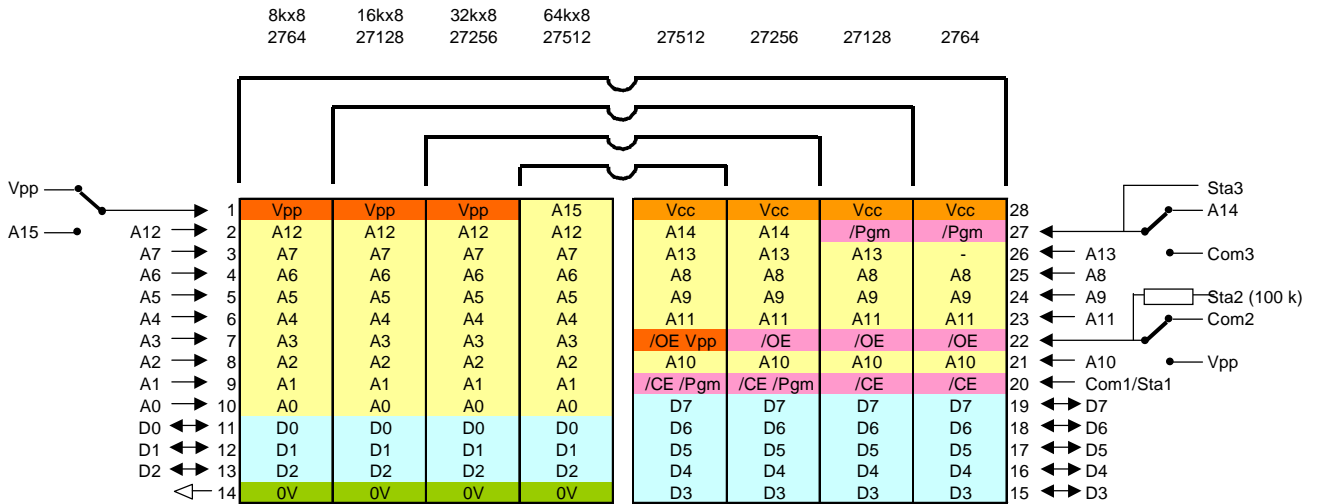
Chaque ligne est de la forme :

:CCAAAARR...ZZCrLf

:	caractère de démarrage de la ligne	
CC	nombre d'octets de données dans la ligne (sans l'adresse, le type, la somme de contrôle)	
AAAA	Offset du premier octet de données de la ligne, par rapport à LBA ou SBA	
RR	Type de ligne	
	00 ligne de données	
	01 ligne de fin	:00000001FF
	02 adresse du segment (Upper Segment Base Address)	:02000002D032F8
	03 execution start address (CS+IP)	:04000003D0000032F8
	04 partie haute de l'adresse sur 32 bits (Upper Linear Base Address)	:02000004D032F8
	05 execution start address (EIP)	:04000005D0000032F8
...	Octets de données	
ZZ	Somme de contrôle. La somme de contrôle est le complément à 2 de la somme des octets à partir du nombre d'octets jusqu'au dernier octet de la zone de données. Autrement dit, la somme de tous les octets à partir du nombre d'octets jusqu'à la somme de contrôle incluse doit être NULLE.	
CrLf	0x0D, 0x0A , Retour chariot, line feed	

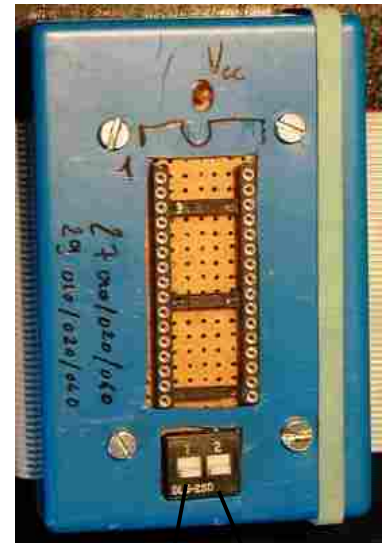
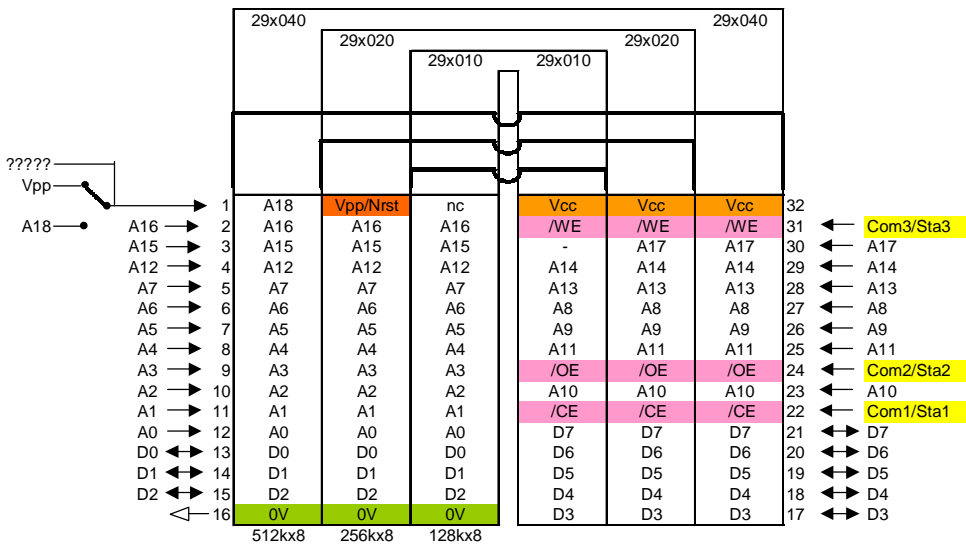
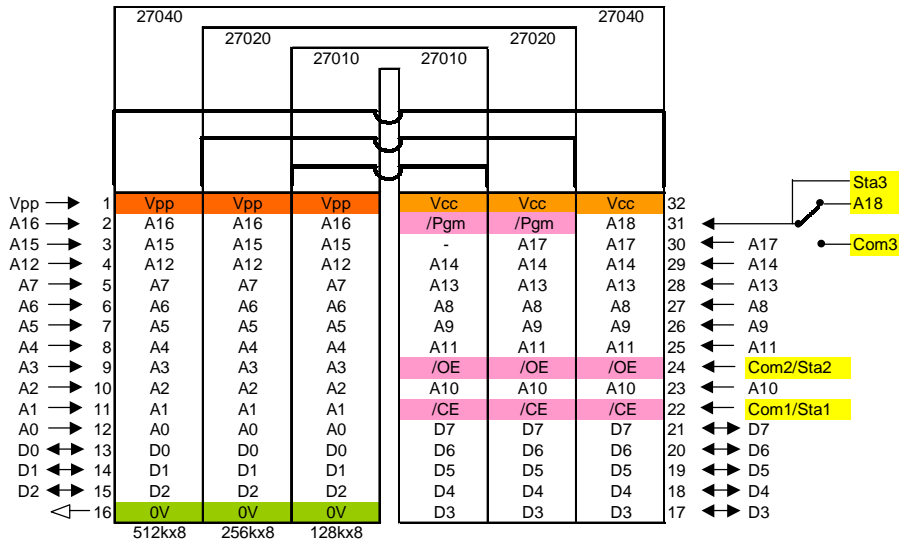
reference : INTEL Hexadecimal Object File Format Specification, janvier 1988, rev A

Interface EPROM 28 broches : 2764, 27128, 27256, 27512



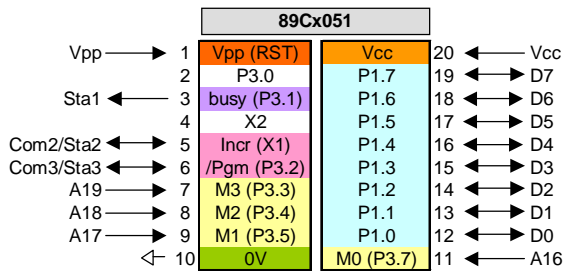
	pin 27	pin 22	pin 1	1	2	1	2
2764	Com3/Sta3	Com2/Sta2	Vpp	down	up	up	
27128	Com3/Sta3	Com2/Sta2	Vpp	down	up	up	
27256	A14/Sta3	Com2/Sta2	Vpp	up	up	up	
27512	A14/Sta3	Vpp	A15	up	down	down	

Interface EPROM 32 broches : 27C010/020/040 ou 29C010/020/040



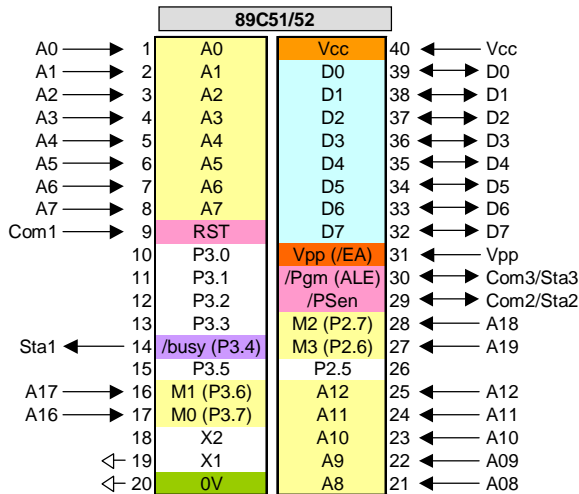
	pin 1	pin 31	1	2
27C010	Vpp	/WE Com3/Sta3	up	down
27C020	Vpp	/WE Com3/Sta3	up	down
27C040	Vpp	A18/Sta3	up	up
29F010	nc	/pgm Com3/Sta3	down	down
29F020	nc ou Vpp/Nrst	/pgm Com3/Sta3	up	down
29F040	A18	/pgm Com3/Sta3	down	down

Interface AT89C1051/2051/4051, AT89C51/52



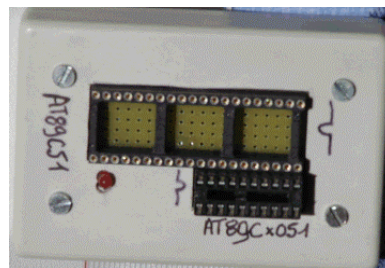
AT89C51/52	/EA	P2.6	P2.7	P3.6	P3.7	ALE	P3.4
AT89C1051/2051	RST	P3.3	P3.4	P3.5	P3.7	P3.2	P3.1

	A19	A18	A17	A16		/program	/busy
	M3	M2	M1	M0			
read signature bit	Vcc	0	0	0	0	1	
read code data	Vcc	0	0	1	1	1	
write code data	Vpp	0	1	1	1	Npulse	Npulse
Chip erase	Vpp	1	0	0	0	10ms Npulse	Npulse
write lock bit 3	Vpp	1	0	1	0	Npulse	Npulse
write lock bit 2	Vpp	1	1	0	0	Npulse	Npulse
write lock bit 1	Vpp	1	1	1	1	Npulse	Npulse



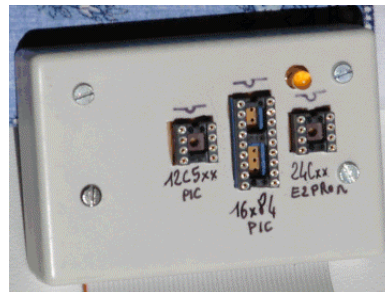
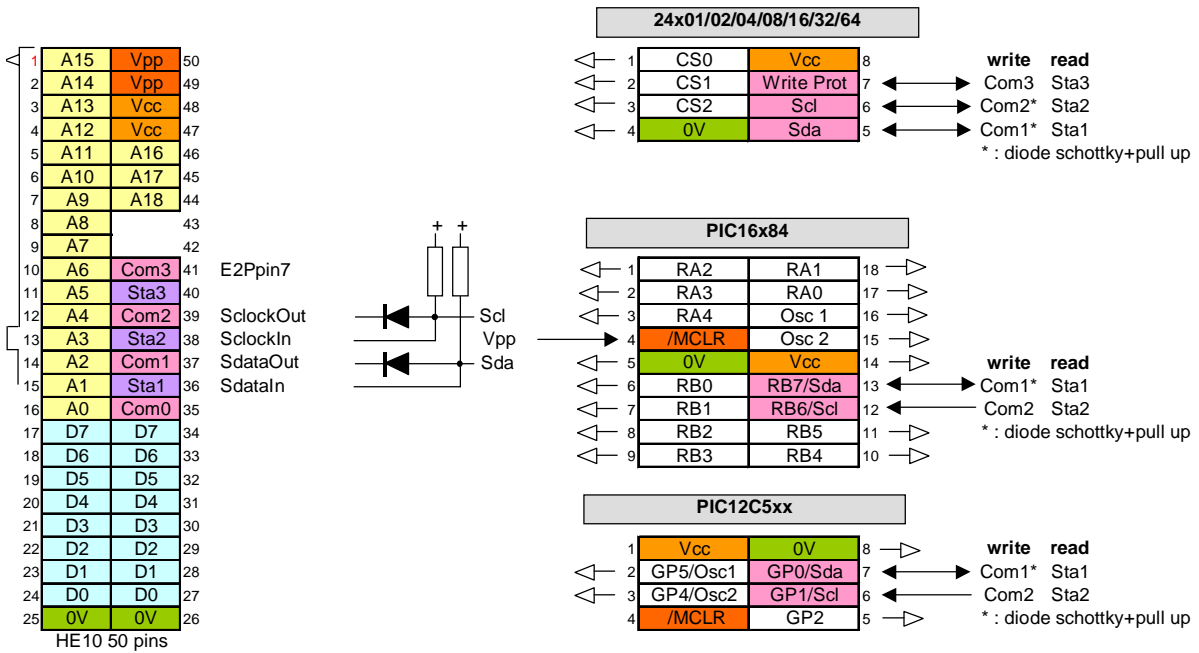
RST à 1 /PSEN à 0

	LB1	LB2	LB3	Protection Type
1	U	U	U	No program lock features
2	P	U	U	MOVX instructions executed from external program memory are disabled from fetching code bytes from internal memory, EA is sampled and latched on reset, and further programming of the Flash memory is disabled.
3	P	P	U	Same as mode 2, but verify is also disabled.
4	P	P	P	Same as mode 3, but external execution is also disabled.



UNIPROG 8/8

Interface PIC16C84, 16F84, 12C508, 12C509 & E2PROM 24Cxxx



Fonctions globales

Lecture circuit ou test virginité	<pre> /OEA=1, /OED=1, OEC=0 /CE=1, /OE=1 puissance = Vcc seulement /OEA=0,OEC=1 /CE = 0, /OE=1 for (i=0;i<taille;i++) { charger adresse(i) ; /OE=0 ; lire donnée ; /OE=1 ; } /CE=1, /OE=1 /OEA=1, /OED=1, OEC=0 puissance = off </pre>
Programmation des données	<pre> /OEA=1, /OED=1, OEC=0 /CE=1, /OE=1, /PGM=1 puissance = Vcc + Vpp /OEA=0, /OED=0, OEC=1 /CE = 0 for (i=0;i<taille;i++) { charger adresse(i) ; charger donnée d(i); /PGM=0 ; attendre t ; /PGM=1 ; /OED=1 ; /OE=0 ; lire & vérifie donnée ; /OE=1 ; /OED=0 ; } /CE=1 /OEA=1, /OED=1, OEC=0 puissance = Vcc seulement puissance = off </pre>
Programmation des codes spéciaux	
Autotest interface sans adaptateur	<pre> puissance = off routine intégrée charger donnée série un coup d'horloge lire donnée série </pre>
Autotest avec adaptateur	

pilotage par un micro controlleur

Signal	Mnémo	Sens	µC	remarque
Commande 0	Com0	In	P1.1	
Commande 1	Com1	In	P1.3	
Commande 2	Com2	In	P1.5	
Commande 3	Com3	In	P1.7	
Autorisation sortie dataW	OED	In	74X259.Qi	
Autorisation sortie adresses	OEA	In	74X259.Qi	Ces signaux peuvent être couplés
Autorisation sortie commandes	OEC	In	74X259.Qi	
Status 1	Sta1	Out	P1.2	
Status 2	Sta2	Out	P1.4	
Status 3	Sta3	Out	P1.6	
Horloge série	Sclock	In	P1.0	
Données série	Sdata	In	P1.1	
Chargement data W	LDW	In	74X259.Qi	
Chargement data R	LDR	In	74X259.Qi	
Shift/load data R	SL	In	SL = /LDR	
Load adresse	LADD	In	74X259.Qi	
Load puissance	LPOW	In	74X259.Qi	
Données série sortie dataW	SDdw	Out	Via Data R	
Données série sortie adresse	SDadd	Out	P3.i via rés	
Données série sortie dataR	SDdr	Out	P3.i via rés	
Données série sortie puissance	SDpow	Out	P3.i via rés	

Utilisation d'un circuit 74x259 pour générer les signaux de contrôle

sorties	mnémo	destination
	LDW	chargement registre données W
	LDR	chargement registre données R
	LADD	chargement registre adresses
	LPOW	chargement registre Vpp et puissance
	/OED	Autorise sortie données W
	/OEA	Autorise sortie adresses
	/OEC	Autorise sortie commande

Séquences d'activation pour un signal statique (autorisation de sortie)

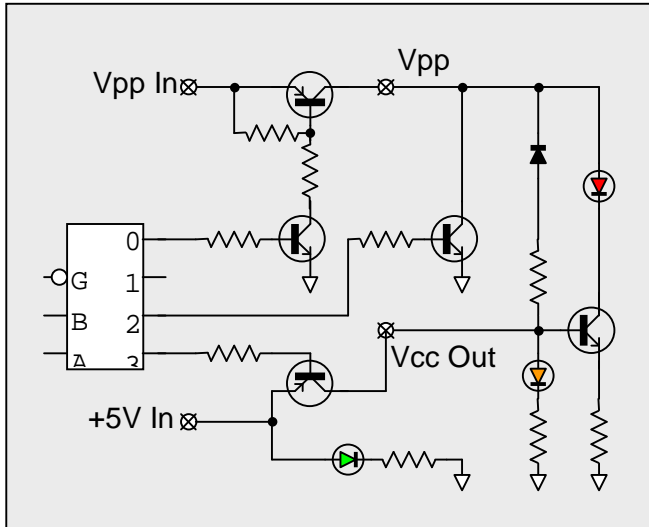
adresse contrôle	donnée 74x259	/G	Sortie Qabc
abc	signal	1	0
abc	signal	0	signal
abc	signal	1	signal

Séquences d'activation pour un signal front (transfert)

adresse contrôle	donnée 74x259	/G	Sortie Qabc
abc	0	1	0
abc	0	0	0
abc	1	0	1
abc	0	0	0
abc	0	1	0

Alimentation du circuit

Avec un décodeur 4555 :



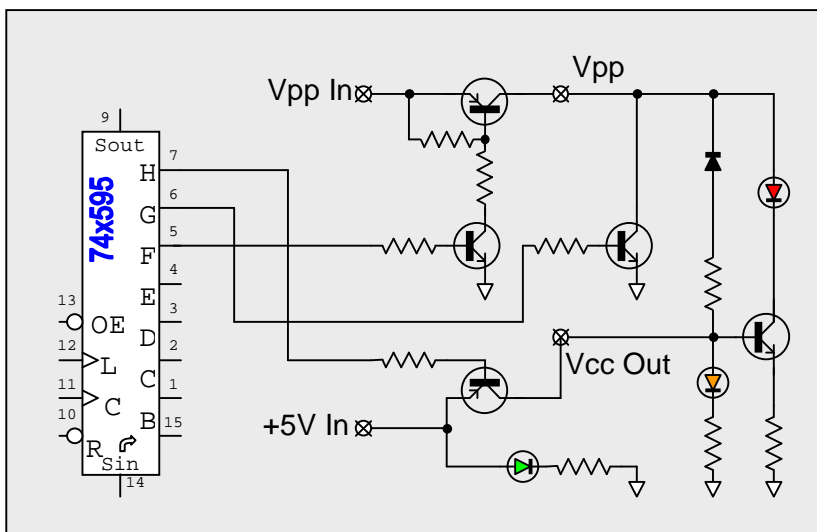
PNP : BD136 pour avoir un V_{sat} faible (typique 0.4 V)

NPN : BC547B

Décodeur : 4555B CMOS

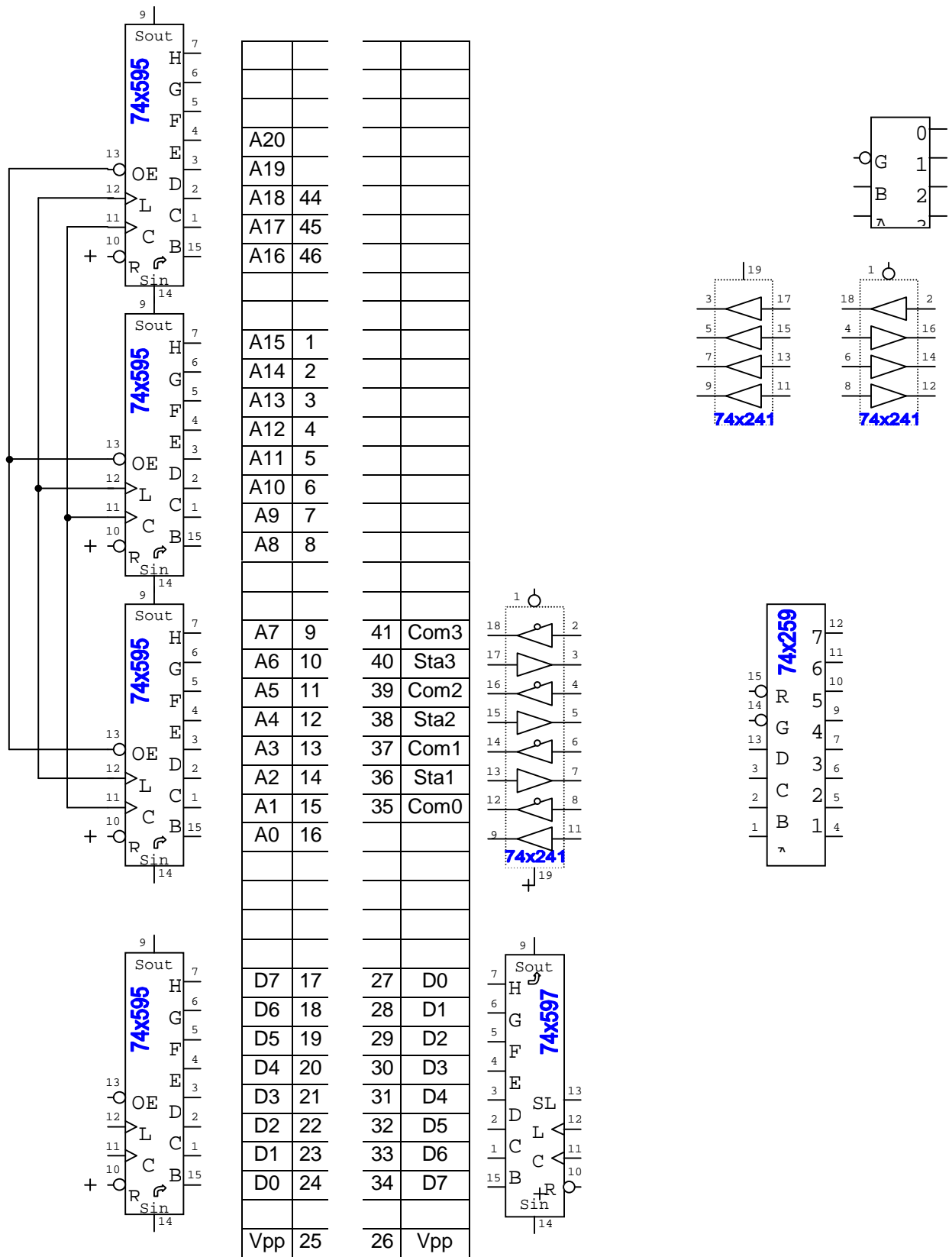
/G	B	A	Vcc Out	Vpp Out
1	X	X	+5 V	+5 V via pull up
0	0	0	+5 V	Vpp
0	0	1	+5 V	+5 V via pull up
0	1	0	+5 V	0 V forcé
0	1	1	OFF	OFF

En direct du registre à décalage :



Q7	Q6	Q5	Vcc Out	Vpp Out
0	0	0	+5 V	+5 V via pull up
0	0	1	+5 V	Vpp
0	1	0	+5 V	0 V forcé
0	1	1	+5 V	court-circuit Vpp
1	0	0	OFF	OFF
1	0	1	OFF	Vpp
1	1	0	OFF	0 V forcé
1	1	1	OFF	court-circuit Vpp

Schéma détaillé



UNIPROG 13/13

Programmation des Flash

512k x 8 256k x 8 128k x 8
29F040 **29C020** **29F010**
29F002
49F002

	Reset	page	DQ6 toggle	DQ7 data polling	Sectors	BootBlock	ID	Write cycle	Erase cycle	8010	8020	8030	8040	90	A0	F0
Winbond	no	128 bytes/page 200 µs/byte	X	X	no	upper 8k or lower 8k	DA 45	10ms	50ms	X	X		X		X	X
Winbond	X	128 bytes/page 200 µs/byte	X	X	no	upper 8k or lower 8k	DA 45									
AMD	X, no	no	X	X	7	lower 16k	01 34			X	X	X		X	X	X
AMD	X, no	no	X	X	7	upper 16k	01 B0			X	X	X		X	X	X
ST	X	no	X	X	7	lower 16k	20 34			X	X	X		X	X	X
ST	X	no	X	X	7	upper 16k	20 B0			X	X	X		X	X	X
ST	X	no	X	X	7	upper 16k	20 B0			X	X	X		X	X	X
Atmel	no	256 bytes/page 150 µs/byte	X	X						X	X		X	X	X	X
Atmel	X, no		X	X	5	lower 16k	1F 07			X	X	X	X	X	X	X
Atmel	X, no		X	X	5	upper 16k	1F 08			X	X	X	X	X	X	X
Amic	X, no		X	X	7	lower 16k	37 0D									
Amic	X, no		X	X	7	upper 16k	37 8C									

ID	commande	séquence
8010	chip erase	5555/AA, 2AAA/55, 5555/80, 5555/AA, 2AAA/55, 5555/10, pause 50ms
8020	disabla data protection	5555/AA, 2AAA/55, 5555/80, 5555/AA, 2AAA/55, 5555/20, pause 10ms
8030	sector erase	5555/AA, 2AAA/55, 5555/80, 5555/AA, 2AAA/55, sector_add/30
8040	BootBlock lockout set	bottom : 5555/AA, 2AAA/55, 5555/80, 5555/AA, 2AAA/55, 5555/40,00000/00, pause 1s top : 5555/AA, 2AAA/55, 5555/80, 5555/AA, 2AAA/55, 5555/40,3FFFF/FF, pause 1s
90	ID entry	5555/AA, 2AAA/55, 5555/90, pause 10µs, read 00000,00001,xxxxx2
A0	enable data protection, program	5555/AA, 2AAA/55, 5555/A0, secteur
F0	ID exit	5555/AA, 2AAA/55, 5555/F0