

FONCTIONNEMENT DU BUS ISA

Un des meilleurs calculateurs à l'heure actuelle est le PC. Pour les robots, mais aussi pour d'autres applications domotiques, ils sont extrêmement performants. Seulement, pour les interfacer avec l'extérieur, ce n'est pas toujours si simple : les ports séries ou parallèles sont lents et ont un nombre de bits limités. C'est pour cette raison que nous allons ici expliquer le fonctionnement d'un bus ISA, et son interfaçage.

Il est possible d'interfacer les Bus ISA AT en 16 bits. Mais nous réaliserons ici l'interfaçage en 8 bits qui est nettement plus simple, et généralement suffisant. Cette fiche technique est valable pour les bus ISA, ISA AT des PC classiques, mais aussi pour les bus PC104.

Fonctionnement

Pour interfacer le bus ISA, nous allons nous intéresser principalement aux données suivantes :

- bus d'adresse (20 bits)
- bus de données (8 bits)
- AEN (1 bit)
- -IOR (1 bit)
- -IOW (1 bit)

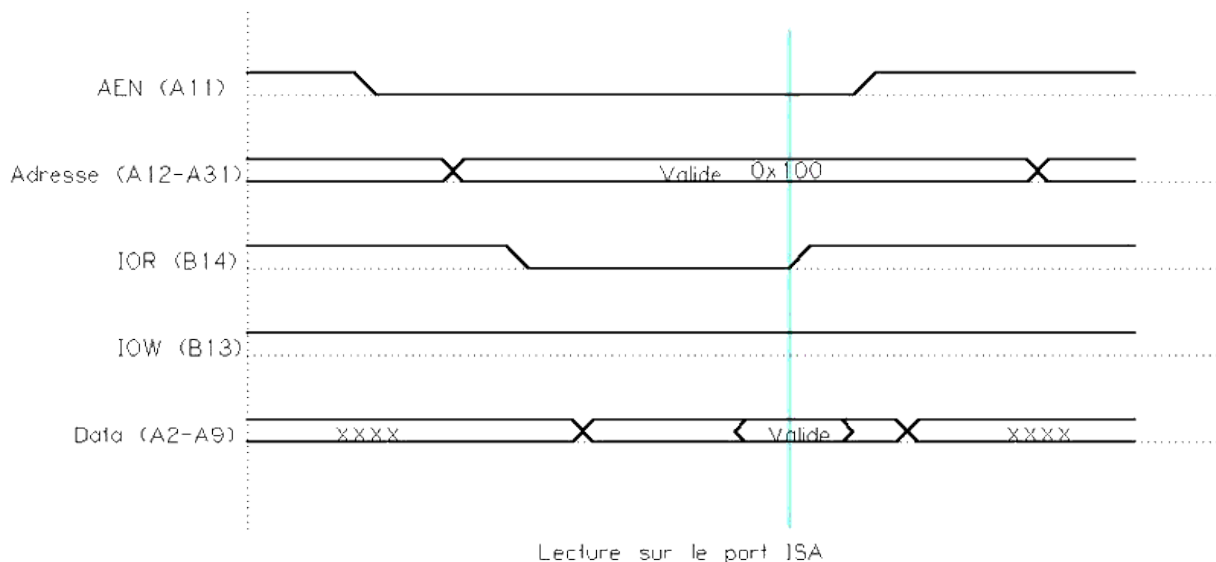
Le bus d'adresse permet de différencier les cartes qui sont mises sur le bus. Le bus de données transporte les informations à communiquer à la carte. Le bit AEN permet d'autoriser la lecture/écriture sur le port. Et enfin -IOR et -IOW servent à différencier la lecture de l'écriture. Prenons l'exemple suivant : nous possédons deux cartes : une carte d'entrée à l'adresse 0x100 et une carte de sortie à l'adresse 0x101.

Exemple 1 : la lecture

Nous souhaitons lire une information à l'adresse 0x100. En C l'instruction appropriée est :

```
Val = inp (0x100);
```

Cette seule instruction va permettre de renvoyer la valeur contenue à l'adresse 0x100 dans la variable entière Val. Voici comment l'instruction va être interprétée par le PC :



AEN va tomber au niveau bas, l'adresse 0x100 va être appliquée sur le bus d'adresse, quand IOR va passer à l'état bas, la valeur appliquée sur le bus de données sera lu par le PC. La carte qui sera interfaçée sur le PC devra garantir de renvoyer la valeur adéquate sur le bus de données chaque fois que les conditions suivantes seront réunies :

- AEN=0,
- A12-A31=Adresse de la carte
- IOR=0
- et IOW=1

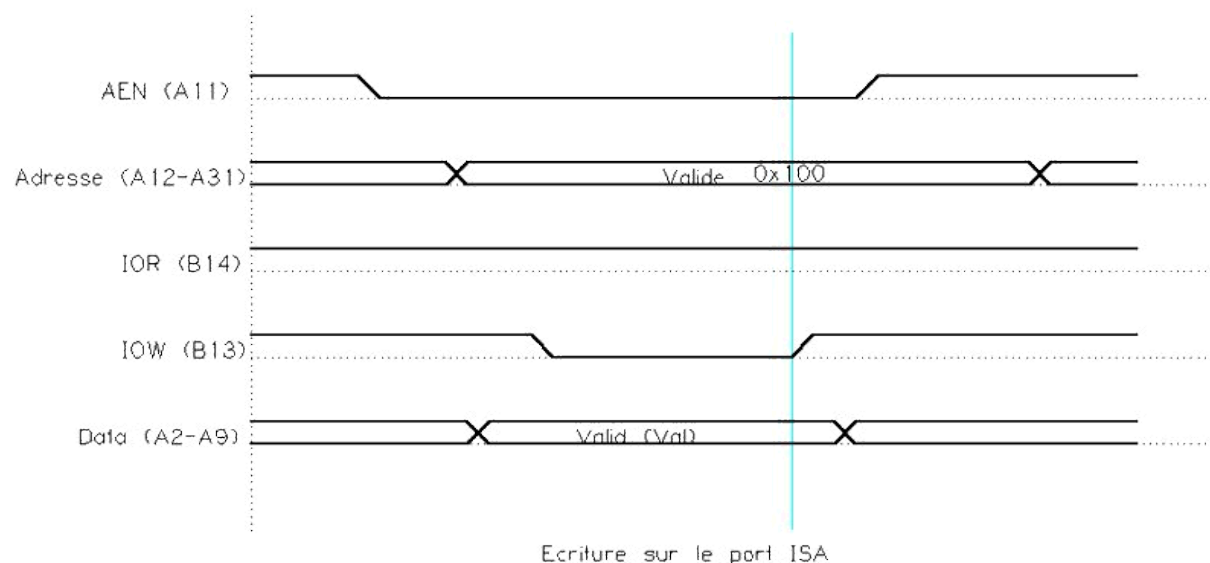
La carte devra appliquer l'état haute impédance le reste du temps afin de garantir qu'il n'y aura pas de conflit avec la mémoire et les autres cartes.

Exemple 2 : l'écriture

Nous souhaitons écrire une information à l'adresse 0x101. En C l'instruction appropriée est :

```
outp (0x100, Val);
```

Cette seule instruction va permettre d'envoyer le contenu de la variable Val dans la carte située à l'adresse 0x101. Voici comment l'instruction va être interprétée par le PC :



AEN va tomber au niveau bas, l'adresse 0x101 va être appliquée sur le bus d'adresse, et la donnée Val sera appliquée sur le bus de données. Quand IOW va passer à zéro, la carte qui sera interfaçée sur le PC devra lire la valeur sur le bus de données. Chaque fois que les conditions suivantes seront réunies, la carte devra lire le bus de données :

- AEN=0,
- A12-A31=Adresse de la carte
- IOR=1
- et IOW=0

Implantation

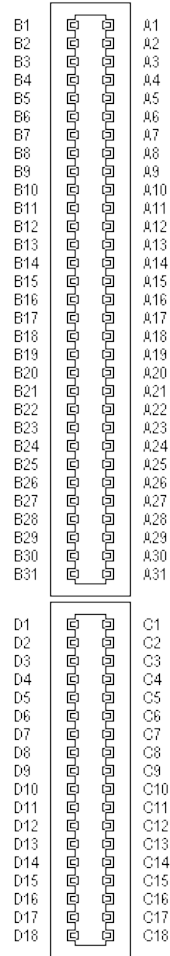
PI	Signal	Description	I/O
A1	-IOCHCK	I/O channel check;	I
A2	D7	Data bit 7	I/O
A3	D6	Data bit 6	I/O
A4	D5	Data bit 5	I/O
A5	D4	Data bit 4	I/O
A6	D3	Data bit 3	I/O
A7	D2	Data bit 2	I/O
A8	D1	Data bit 1	I/O
A9	D0	Data bit 0	I/O
A10	-IOCHRDY	Channel ready;	I
A11	AEN	Address enable; active high	O
A12	A19	Address bft 19	O
A13	A18	Address bft 18	O
A14	A17	Address bft 17	O
A15	A16	Address bft 16	O
A16	A15	Address bft 15	O
A17	A14	Address bft 14	O
A18	A13	Address bft 13	O
A19	A12	Address bft 12	O
A20	A11	Address bft 11	O
A21	A10	Address bft 10	O
A22	A9	Address bft 9	O
A23	A8	Address bft 8	O
A24	A7	Address bft 7	O
A25	A6	Address bft 6	O
A26	A5	Address bft 5	O
A27	A4	Address bft 4	O
A28	A3	Address bft 3	O

PI	Signal	Description	I/O
B1	OV	Ground	Pow
B2	RESET DRV	Active high Io reset	O
B3	+5Vdc	+5Vdc	Pow
B4	IRQ9	Interrupt request 9	I
B5	-5Vdc	-5Vdc	Pow
B6	DRQ2	DMA request 2	I
B7	-12Vdc	-12Vdc	Pow
B8	-CARD	SLCTD Card selected;	I
B9	+12Vdc	+12Vdc	Pow
B10	OV	Ground	Pow
B11	-MEMW	Memory write	O
B12	-MEMR	Memory read	O
B13	-IOW	I/O write	O
B14	-IOR	I/O read	O
B15	-DACK3	DMA acknowledge 3	O
B16	DRQ3	DMA request 3	I
B17	-DACK1	DMA acknowledge 1	O
B18	DRQ1	DMA request 1	I
B19	-REFRESH	Refresh	I/O
B20	CLOCK	System clock 6 or 8MHz	O
B21	IRQ7	Interrupt request 7	I
B22	IRQ6	Interrupt request 6	I
B23	IRQ5	Interrupt request 5	I
B24	IRQ4	Interrupt request 4	I
B25	IRQ3	Interrupt request 3	I
B26	-DACK2	DMA acknowledge 2	O
B27	T/C	Terminal count: pulses high	O
B28	ALE	Address latch enable	O

A29	A2	Address bft 2	O
A30	A1	Address bft 1	O
A31	A0	Address bft 0	O

B29	+5Vdc	+5Vdc	Pow
B30	OSC	HS clock 14.31818Mhz	O
B31	0V	Ground	Pow

Slot vue de dessus



PC104 BUS ISA

IN.....IO CHCHK	A1	□	B1	Ground
I/O.....SD7	A2	□	B2	ResetOUT
I/O.....SD6	A3	□	B3	+5V
I/O.....SD5	A4	□	B4	IRQ2.....IN
I/O.....SD4	A5	□	B5	-5V
I/O.....SD3	A6	□	B6	DRQ2.....IN
I/O.....SD2	A7	□	B7	-12V
I/O.....SD1	A8	□	B8	N/A
I/O.....SD0	A9	□	B9	+12V
IN.....IO CRDY	A10	□	B10	N/A
I/O.....AEN	A11	□	B11	-SMEMW.....I/O
I/O.....SA19	A12	□	B12	-SMEMR.....I/O
I/O.....SA18	A13	□	B13	-IOW.....I/O
I/O.....SA17	A14	□	B14	-IOR.....I/O
I/O.....SA16	A15	□	B15	-DACK3.....OUT
I/O.....SA15	A16	□	B16	DRQ3.....IN
I/O.....SA14	A17	□	B17	-DACK1.....OUT
I/O.....SA13	A18	□	B18	DRQ1.....IN
I/O.....SA12	A19	□	B19	-Refresh.....I/O
I/O.....SA11	A20	□	B20	CLK.....OUT
I/O.....SA10	A21	□	B21	IRQ7.....IN
I/O.....SA9	A22	□	B22	IRQ6.....IN
I/O.....SA8	A23	□	B23	IRQ5.....IN
I/O.....SA7	A24	□	B24	IRQ4.....IN
I/O.....SA6	A25	□	B25	IRQ3.....IN
I/O.....SA5	A26	□	B26	-DACK2.....OUT
I/O.....SA4	A27	□	B27	T/C.....OUT
I/O.....SA3	A28	□	B28	BALE.....OUT
I/O.....SA2	A29	□	B29	+5V
I/O.....SA1	A30	□	B30	OSC.....OUT
I/O.....SA0	A31	□	B31	Ground
Ground	A32	□	B32	Ground

Quelques liens

- [Une carte ISA réalisée par les Fribottes](#)
- [Une carte ISA réalisée par l'ENS Lyon](#)
- [Type 1, un robot avec une carte PC104](#)
- [Les cartes d'extensions](#)

Contact

Pour toutes questions envoyer moi un mail: [Sinclair](#)

ICQ# : [144345434](#)

