8259

Programmable Interrupt Controller (PIC)

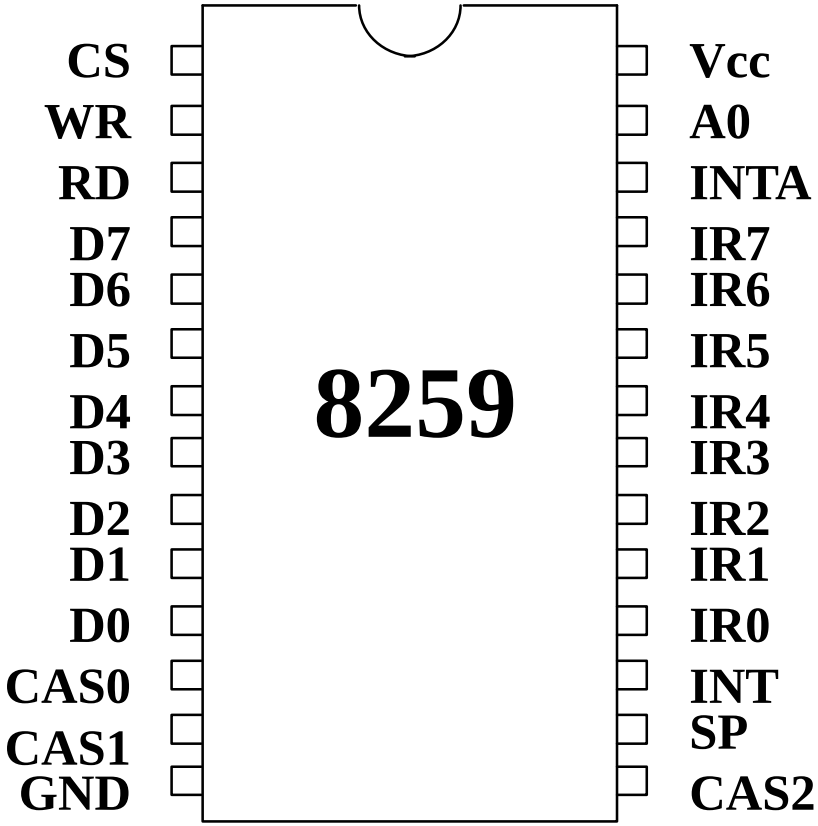
M. Rebaudengo - M. Sonza Reorda

Politecnico di Torino
Dip. di Automatica e Informatica

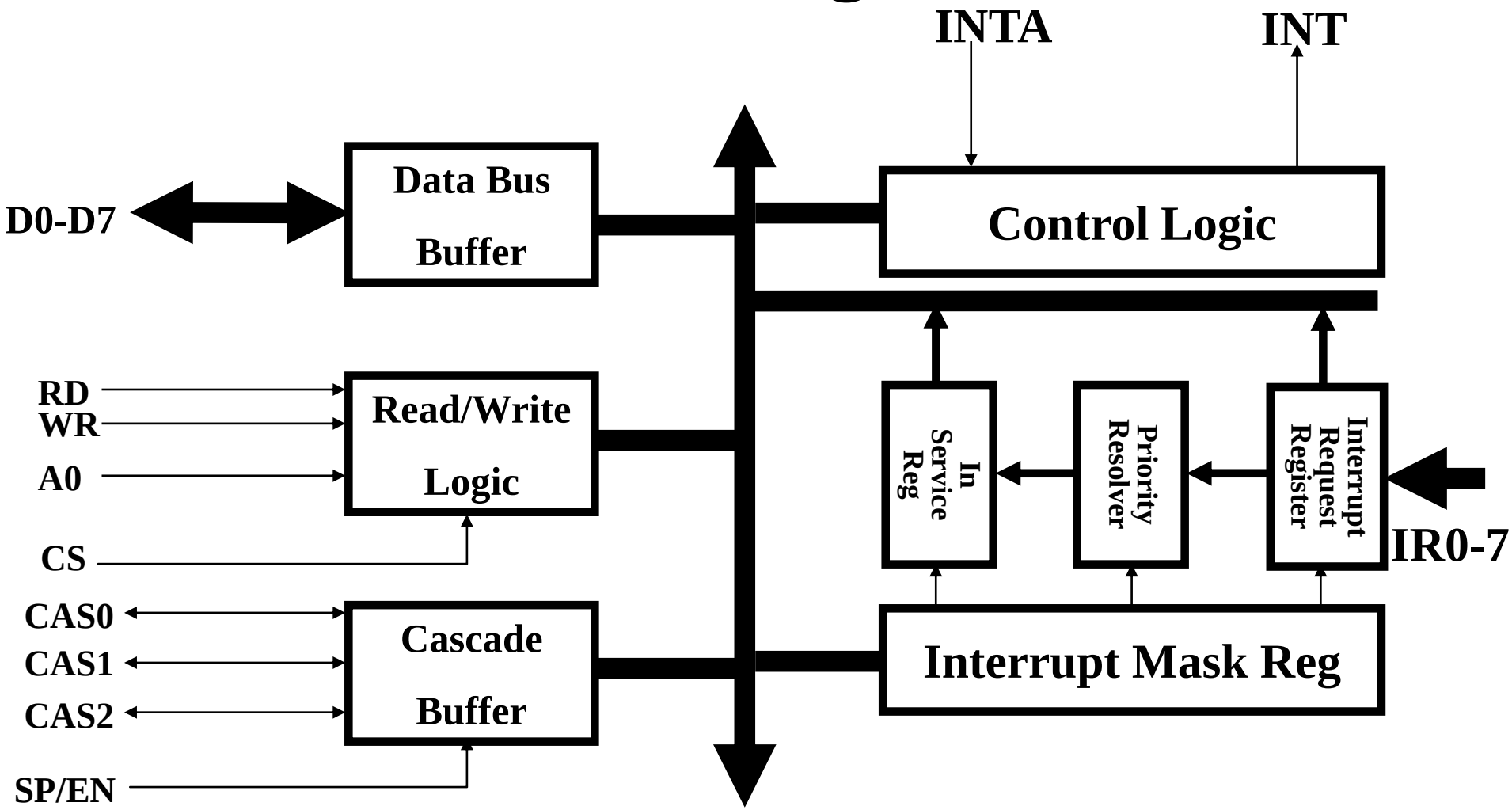
Programmable Interrupt Controller

- **The 8259 was designed to minimize software and latency in the management of multiple interrupt levels with different priority.**
- **The device is a LSI chip inside a 28-pin DIP.**
- **It can manage up to 8 levels of vectored interrupts.**
- **It can be interconnected with other 8259 slaves to handle up to 64 priority levels.**
- **Different priority mechanisms are available for every interrupt level.**
- **The circuitry of 8259 is static (without clock in input).**

The chip


CS
WR
RD
D7
D6
D5
D4
D3
D2
D1
D0
CAS0
CAS1
GND
8259
Vcc
A0
INTA
IR7
IR6
IR5
IR4
IR3
IR2
IR1
IR0
INT
SP
CAS2

Block diagram



Interrupt sequence

- One or more interrupt request lines are raised high. If they are not masked, the corresponding bits in IRR register are set.
- The 8259 holds all interrupt requests and sends an INT signal to the CPU.
- The CPU acknowledges the request and responds with a first INTA signal.
- The request of highest priority is selected by setting the corresponding bit in the ISR register and by resetting the corresponding bit in the IRR register.

Interrupt sequence

- **The CPU sends a second INTA signal.**
- **The 8259 communicates on the data bus the code (1 byte) of the device that requested the interrupt.**
- **The interrupt cycle is completed by resetting the bit in ISR. In AEOI (*automatic end of interrupt*) mode, the bit is automatically resetted. Otherwise, the interrupt cycle needs to be completed by means of an EOI (*end of interrupt*) instruction.**
- **The 8086 activates the interrupt service routine (ISR).**

Programming the 8259

The 8259 is programmed by means of two types of control words:

- ***Initialization Command Words* (ICWs)**
- ***Operation Command Words* (OCWs).**

The order of the ICWs is fixed and they normally are sent only once during the initialization. The OCWs can be sent individually in every phase of the program.

ICWs

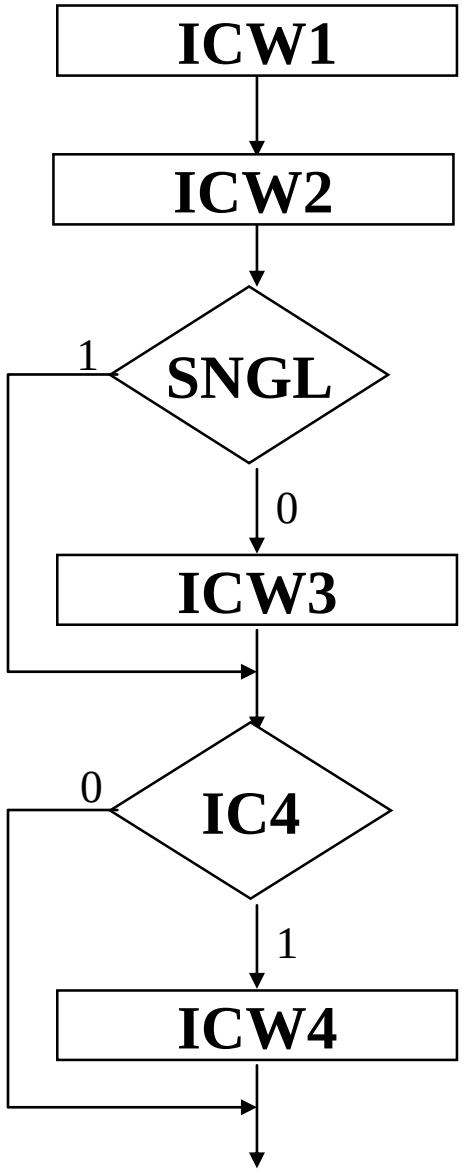
The initialization of the 8259 occurs with a sequence of control words.

This sequence can be recognized because the first data (ICW1) has

- **address signal A0 = 0**
- **bit D4 = 1.**

When the CPU sends a sequence of ICW, the IMR mask register is resetted.

ICWs



ICW1

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	X	X	X	1	LTIM	X	SNGL	IC4

- **LTIM (Level Triggered Mode)**
- **SNGL: Single or Cascade mode**
- **IC4: presence of ICW4**
- **D4 = 1**
- **X: bit used only for CPU of 8080/8085 family.**

LTIM

0	1
edge triggered	level triggered

IC4

0	1
ICW4 is missing	ICW4 is present

SNGL

0	1
cascade mode	single 8259

ICW2

A0	D7	D6	D5	D4	D3	D2	D1	D0
1	T7	T6	T5	T4	T3	X	X	X

With ICW2, the CPU determines the kind of interrupt corrisponding to the 8 signals of interrupt request:

- **T3-T7: 5 upper bits of the address of the interrupt vector**
- **X: bit used only for CPU of 8080/8085 family.**

ICW3

Master device

A0	D7	D6	D5	D4	D3	D2	D1	D0
1	S7	S6	S5	S4	S3	S2	S1	S0

Every bit of ICW3 indicates if the corresponding IR signal is an 8259 slave (bit set to 1) or a normal peripheral (bit set to 0).

Slave device

A0	D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	ID2	ID1	ID0

It informs the slave which IR input pf the master is connected to it.

ICW4

A0	D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	SFNM	BUF	M/S	AEOI	μPM

- **SFNM: Special fully nested mode.**
- **BUF & M/S: Buffer Mode.**
- **AEOI: Automatic end of interrupt.**
- **μPM: Microprocessor mode.**

μPM

0	1
8080/85 mode	8086 mode

AEOI

0	1
Normal EOI	Automatic End Of Interrupt

SFNM

0	1
No Special Fully Nested Mode	Special Fully Nested Mode

Automatic End of Interrupt Mode

- **If the bit AEOI of ICW4 is set to 1, the 8259 operates in AEOI mode.**
- **In AEOI mode, the 8259 resets automaticaly the ISR register after the rise of the second INTA signal.**

End of Interrupt

- **If the AEOI bit of ICW4 is 0, then an explicit command is required to reset the IS bit in ISR register.**
- **This command should be the last instruction before IRET.**
- **In a cascade of 8259, two EOI commands are needed, one for the master and the other one for the slave.**
- **There are two kinds of EOI:**
  - ***Specific End of Interrupt***
  - ***Non Specific End of Interrupt.***

Non Specific EOI

- **It is used with *fully nested mode*, i.e., with static priority.**
- **With a non specific EOI, the bit with highest priority in the ISR register, which corresponds to the last served interrupt level, is reset.**
- **A *Non Specific EOI* is sent with OCW2 (R = 0, SL = 0, EOI = 1).**

Specific EOI

- **Without the *fully nested* mode, the priority changes, so the 8259 can not recognize the last enabled interrupt level.**
- **The EOI command needs to specify the interrupt level to reset in the ISR register.**
- **The *Specific EOI* is sent with OCW2 (R = 0, SL = 1, EOI = 1 and bit L_0-L_2 with the value of level to reset).**

Fully Nested Mode

- **The interrupt requests are ordered according to the priority level, from 0 to 7. Level 0 has the highest priority.**
- **When the CPU enables an interrupt request (with the first INTA pulse), the 8259 computes the request with highest priority; the corresponding interrupt code is put on the Data Bus and the corresponding bit of ISR register is set.**
- **The bit in ISR is set until the CPU sends an EOI command immediately before returning from the interrupt service routine. Instead, if AEOI bit is set, then the bit in ISR is reset automaticaly after the rise of the second INTA signal.**
- **While the bit in ISR is set, all subsequent requests with lower priority are disabled. Only requests with higher priority generates an interrupt request**

Special Fully Nested Mode

- **This mode is used with an 8259 cascade, in order to keep priority within every slave.**
- **When an interrupt request is being served, this slave is not blocked from the master, but eventual requests with higher priority from the same slave will be recognized by the master.**
- **When exiting from the interrupt service routine, before sending the EOI command to the master, the ISR register should be read (after sending a nonspecific EOI). If the ISR register is zero, then the nonspecific EOI can be sent to the master.**

Buffered mode

BUF	0	1	1
M/S	X	0	1
	No buffered mode	Buffered mode slave	Buffered mode master

- In *buffered* mode, the SP/EN pin is an output pin that enables the tranceiver when the 8259 executes a write on the Data Bus.
- In *non buffered* mode, the bit M/S indicates wether the chip is master or slave.

OCW1

A0	D7	D6	D5	D4	D3	D2	D1	D0
1	M7	M6	M5	M4	M3	M2	M1	M0

OCW1 writes the IMR register.

- **Every bit in OCW1 corresponds to a bit in IMR.**
- **If bit M_i is set to 1, then the bit IMR_i is set to 1, so the interrupt channel IR_i is masked.**

OCW2

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	R	SL	EOI	0	0	L2	L1	L0

- The R, SL and EOI bits determine the management of the priority rotation and of the end of interrupt.
- The L2-L0 bits specify a particular interrupt channel.

OCW2

R	SL	EOI	
0	0	1	Non Specific EOI command
0	1	1	Specific EOI command
1	0	1	Rotate on Non Specific EOI command
1	0	0	Rotate in AEOI mode (set)
0	0	0	Rotate in AEOI mode (clear)
1	1	1	Rotate on Specific EOI command
1	1	0	Set Priority Command
0	1	0	No operation

OCW3

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	ESMM	SMM	0	1	P	RR	RIS

- **The ESMM and SMM bits set or reset the Special Mask Mode.**
- **The RR e RIS bits handle the readings of internal registers (IRR and ISR).**
- **The P bit sets or resets the Poll Command.**

Read Register Command

RIS	X	0	1
RR	0	1	1
	No action	Read IRR	Read ISR

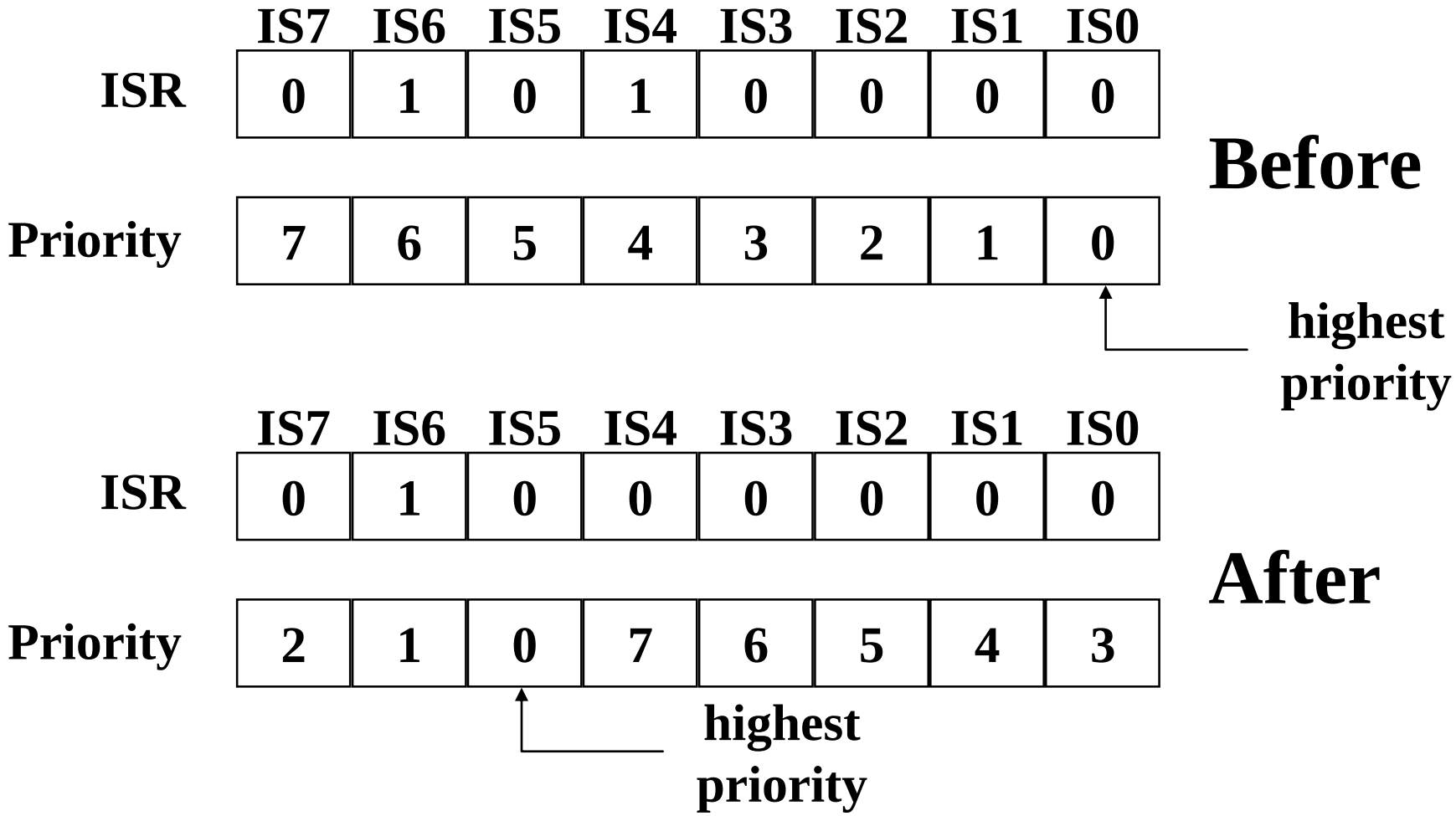
Special Mask Mode

SMM	X	0	1
ESMM	0	1	1
	No action	Reset Special Mask	Set Special Mask

Automatic Rotation mode

- **In many applications, it can happen that different I/O devices have the same priority.**
- **In these cases, rotating priority levels should be preferred with respect to static priority.**
- **When a device is served, it receives the lowest priority level.**

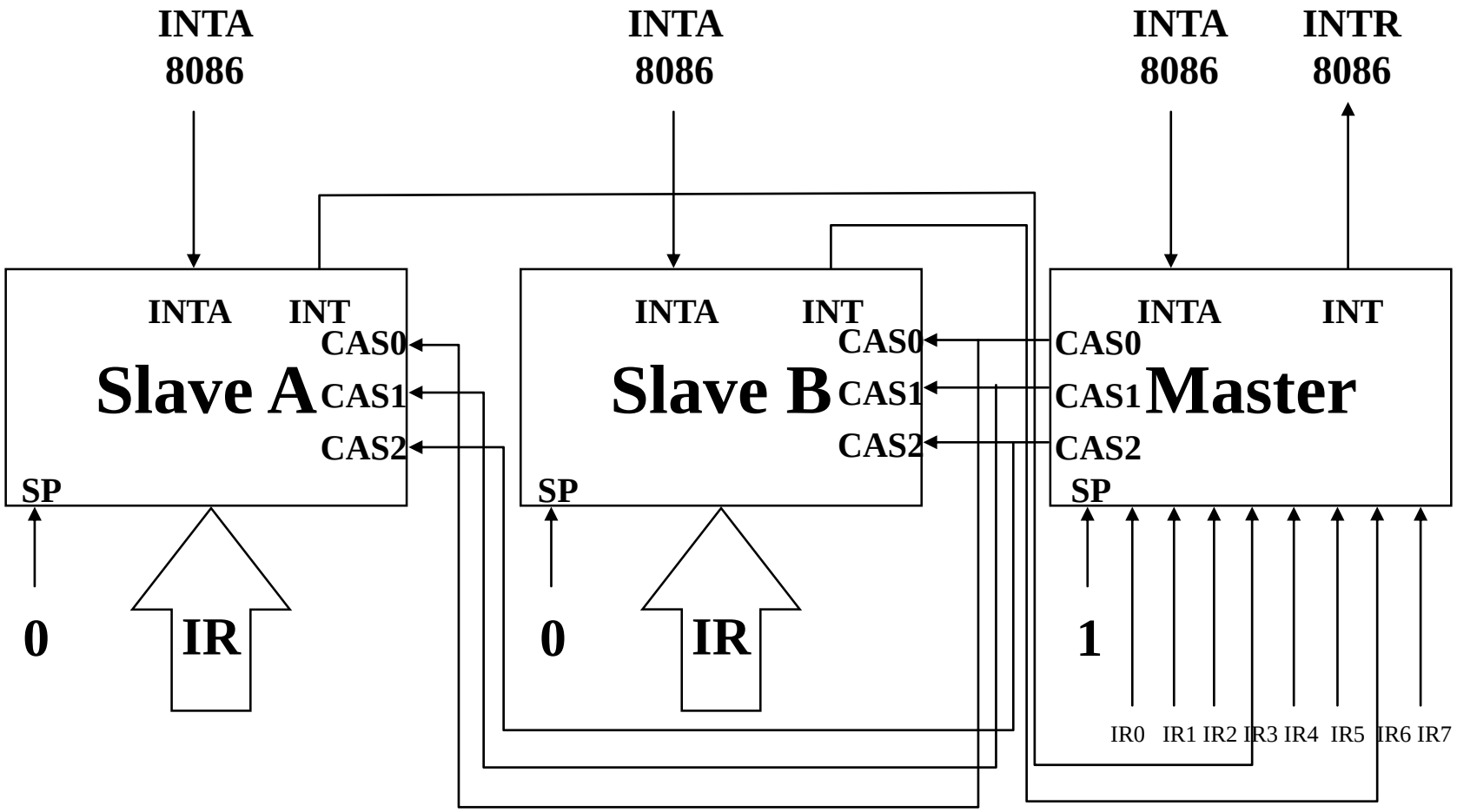
Rotating priority



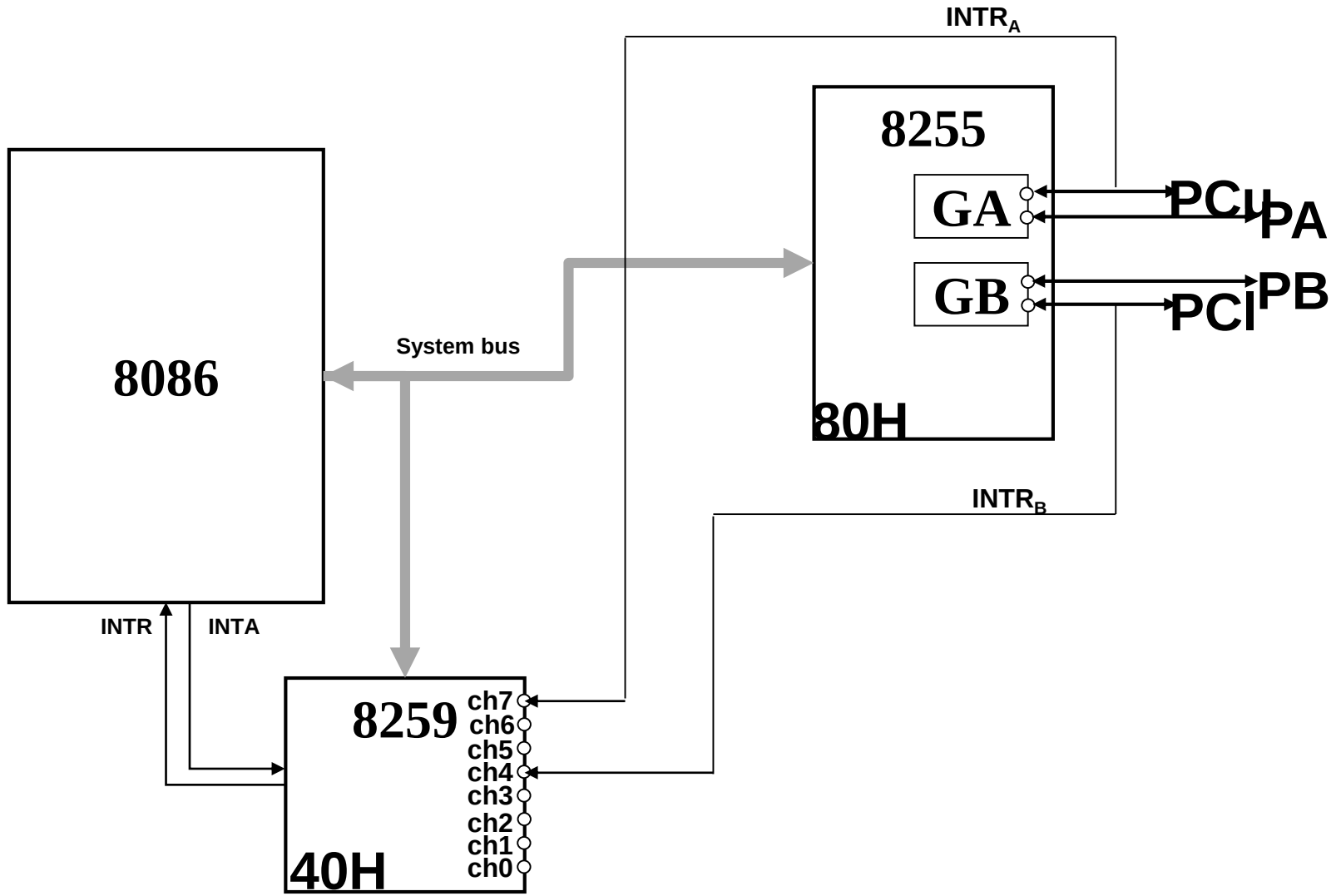
Cascade of 8259

- **The system of interrupt controller can be expanded in order to handle up to 64 interrupt levels, by using one master and 8 slave.**
- **The master controls the slaves by means of 3 lines of CAS. These lines operates like a chip select for the slaves during the INTA sequence.**
- **Every 8259 needs to be properly programmed.**
- **The EOI commands for the interrupt levels originated by the slaves need to be repeated two times: one for the master and the other for the slave.**

Example


INTA
8086
INTA
8086
INTA
8086
INTR
8086
INTA
INT
CAS0
CAS1
CAS2
Slave A
SP
0
IR
INTA
INT
CAS0
CAS1
CAS2
Slave B
SP
0
IR
INTA
INT
CAS0
CAS1
CAS2
Master
SP
1
IR0
IR1
IR2
IR3
IR4
IR5
IR6
IR7

Emulated system with EMU8086



Example

- **Configure the 8255 in mode 1 for groups A and B, with group A in input and group B in output.**
- **Write the interrupt service routines such that:**
  - **When an interrupt request from group A is received, port A is read**
  - **If the byte corresponds to the ASCII code of a decimal digit $n > 0$, write on port B the value of this digit**
  - **Then, write on port B the values n-1, n-2, …, 1**
  - **Assume that a new character does not arrive on port A until the writing of all values on port B is in progress.**