

# Hardware-Software Interfaces

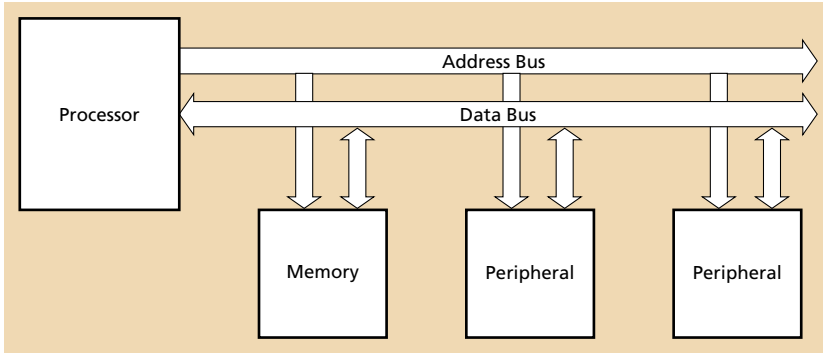
CSEE W4840

Prof. Stephen A. Edwards

Columbia University

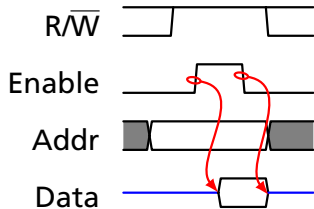
Spring 2026

# Processor System Block Diagram

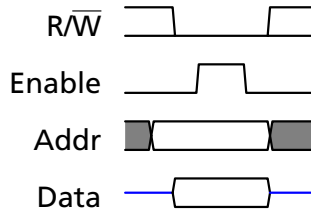


# Simple Bus Timing

## Read Cycle

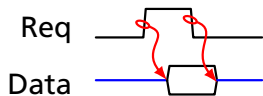


## Write Cycle

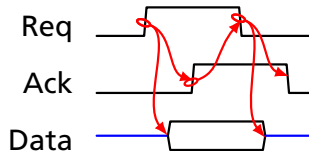


# Strobe vs. Handshake

**Strobe**



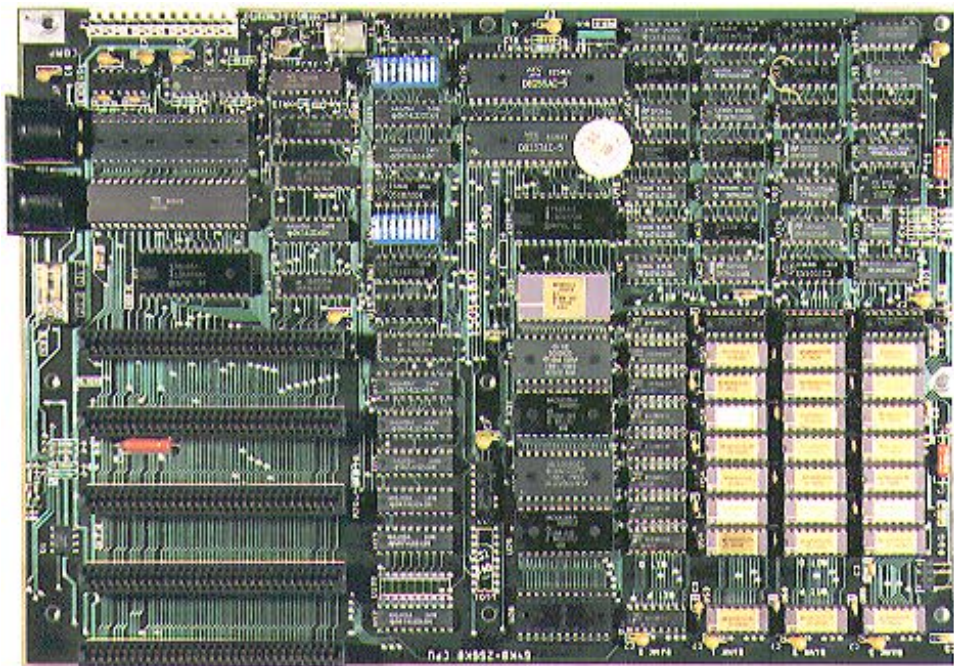
**Handshake**



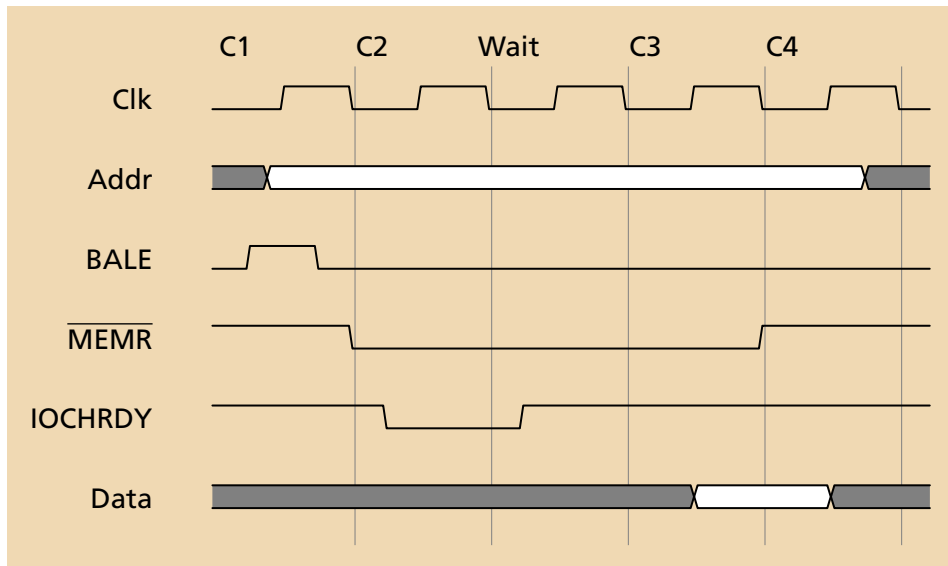
1982:

The  
IBM  
PC

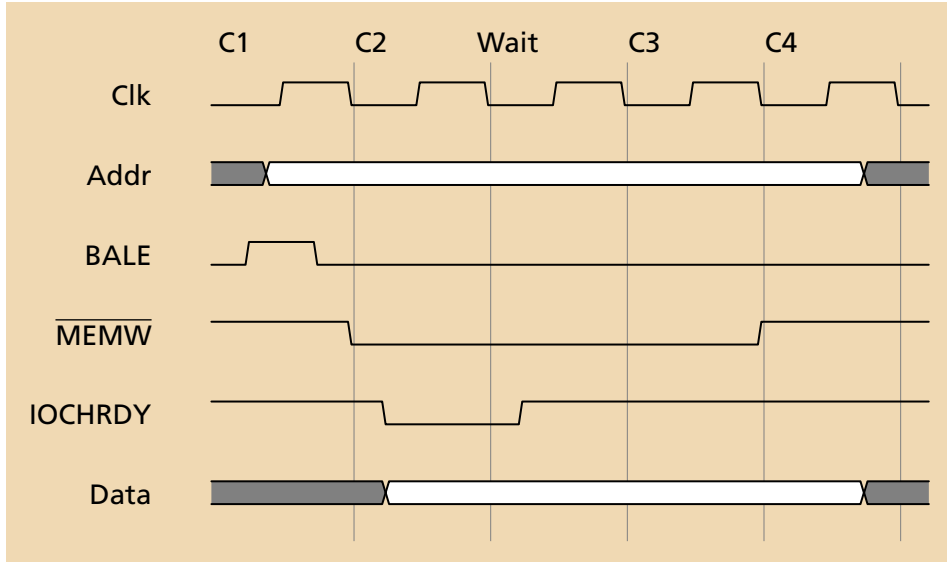
Intel  
8088  
4.77  
MHz  
8-bit  
ISA  
bus



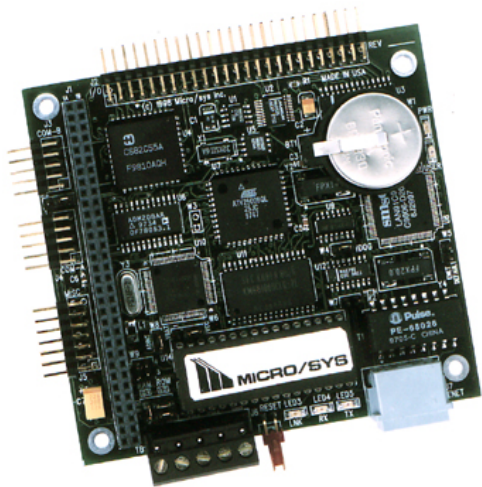
# The ISA Bus: Memory Read



# The ISA Bus: Memory Write



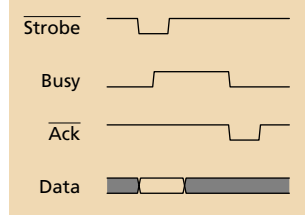
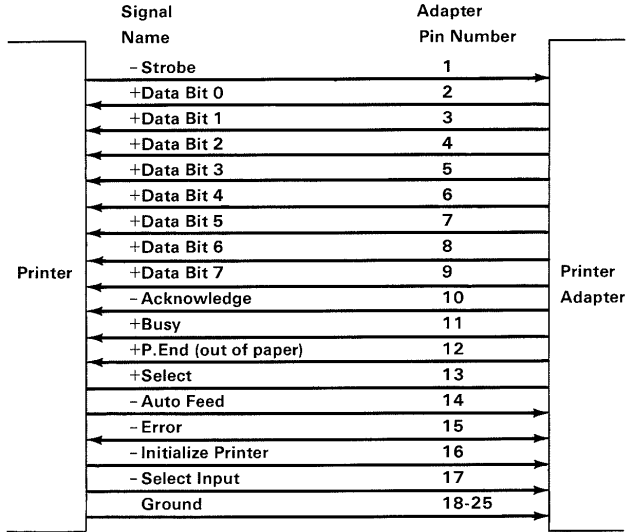
# The PC/104 Form Factor: ISA Lives



# Memory-Mapped I/O

- ▶ To a processor, everything is memory.
- ▶ Peripherals appear as magical memory locations.
- ▶ Status registers: when read, report state of peripheral
- ▶ Control registers: when written, change state of peripheral

# Typical Peripheral: The PC Parallel Port



|                |                |                |                |                           |                |                |                |
|----------------|----------------|----------------|----------------|---------------------------|----------------|----------------|----------------|
|                |                |                |                | <b>Printer Adapter</b>    |                |                |                |
|                |                |                |                | Output to address hex 378 |                |                |                |
| Bit 7<br>Pin 9 | Bit 6<br>Pin 8 | Bit 5<br>Pin 7 | Bit 4<br>Pin 6 | Bit 3<br>Pin 5            | Bit 2<br>Pin 4 | Bit 1<br>Pin 3 | Bit 0<br>Pin 2 |

|        |        |        |        |                            |       |       |       |
|--------|--------|--------|--------|----------------------------|-------|-------|-------|
|        |        |        |        | <b>Printer Adapter</b>     |       |       |       |
|        |        |        |        | Input from address hex 379 |       |       |       |
| Bit 7  | Bit 6  | Bit 5  | Bit 4  | Bit 3                      | Bit 2 | Bit 1 | Bit 0 |
| Pin 11 | Pin 10 | Pin 12 | Pin 13 | Pin 15                     | —     | —     | —     |

|  |                            |                                         |                     |                                         |                                        |
|--|----------------------------|-----------------------------------------|---------------------|-----------------------------------------|----------------------------------------|
|  |                            | Printer Adapter                         |                     |                                         |                                        |
|  |                            | Output to address hex 37A               |                     |                                         |                                        |
|  | Bit 4<br><br>IRQ<br>Enable | $\overline{\text{Bit 3}}$<br><br>Pin 17 | Bit 2<br><br>Pin 16 | $\overline{\text{Bit 1}}$<br><br>Pin 14 | $\overline{\text{Bit 0}}$<br><br>Pin 1 |

IBM Personal Computer  
Technical Reference  
Manual, 1983

Table 2-2: Parallel port bits, arranged by register.

| Data Register (Base Address) |            |             |        |                        |                 |
|------------------------------|------------|-------------|--------|------------------------|-----------------|
| Bit                          | Pin: D-sub | Signal Name | Source | Inverted at connector? | Pin: Centronics |
| 0                            | 2          | Data bit 0  | PC     | no                     | 2               |
| 1                            | 3          | Data bit 1  | PC     | no                     | 3               |
| 2                            | 4          | Data bit 2  | PC     | no                     | 4               |
| 3                            | 5          | Data bit 3  | PC     | no                     | 5               |
| 4                            | 6          | Data bit 4  | PC     | no                     | 6               |
| 5                            | 7          | Data bit 5  | PC     | no                     | 7               |
| 6                            | 8          | Data bit 6  | PC     | no                     | 8               |
| 7                            | 9          | Data bit 7  | PC     | no                     | 9               |

Some Data ports are bidirectional. (See Control register, bit 5 below.)

| Status Register (Base Address +1) |            |                 |            |                        |                 |
|-----------------------------------|------------|-----------------|------------|------------------------|-----------------|
| Bit                               | Pin: D-sub | Signal Name     | Source     | Inverted at connector? | Pin: Centronics |
| 3                                 | 15         | nError (nFault) | Peripheral | no                     | 32              |
| 4                                 | 13         | Select          | Peripheral | no                     | 13              |
| 5                                 | 12         | PaperEnd        | Peripheral | no                     | 12              |
| 6                                 | 10         | nAck            | Peripheral | no                     | 10              |
| 7                                 | 11         | Busy            | Peripheral | yes                    | 11              |

Additional bits not available at the connector:

0: may indicate timeout (1=timeout).

1, 2: unused.

| Control Register (Base Address +2) |            |             |                 |                        |                 |
|------------------------------------|------------|-------------|-----------------|------------------------|-----------------|
| Bit                                | Pin: D-sub | Signal Name | Source          | Inverted at connector? | Pin: Centronics |
| 0                                  | 1          | nStrobe     | PC <sup>1</sup> | yes                    | 1               |
| 1                                  | 14         | nAutoLF     | PC <sup>1</sup> | yes                    | 14              |
| 2                                  | 16         | nInit       | PC <sup>1</sup> | no                     | 31              |
| 3                                  | 17         | nSelectIn   | PC <sup>1</sup> | yes                    | 36              |

<sup>1</sup>When high, PC can read external input (SPP only).

Additional bits not available at the connector:

4: Interrupt enable. 1=IRQs pass from *nAck* to system's interrupt controller. 0=IRQs do not pass to interrupt controller.

5: Direction control for bidirectional Data ports. 0=outputs enabled. 1=outputs disabled; Data port can read external logic voltages.

6,7: unused

Jan Axelson, *Parallel Port Complete*, 2002

|              |                    |
|--------------|--------------------|
| DATA PORT    | BASE ADDRESS + 00H |
| STATUS PORT  | BASE ADDRESS + 01H |
| CONTROL PORT | BASE ADDRESS + 02H |

The bit map of these registers is:

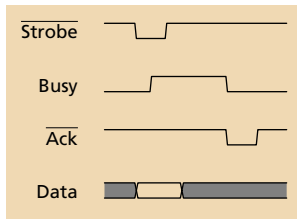
|              | <b>D0</b> | <b>D1</b> | <b>D2</b> | <b>D3</b> | <b>D4</b> | <b>D5</b> | <b>D6</b> | <b>D7</b> |
|--------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| DATA PORT    | PD0       | PD1       | PD2       | PD3       | PD4       | PD5       | PD6       | PD7       |
| STATUS PORT  | TMOUT     | 0         | 0         | nERR      | SLCT      | PE        | nACK      | nBUSY     |
| CONTROL PORT | STROBE    | AUTOFD    | nINIT     | SLC       | IRQE      | PCD       | 0         | 0         |

SMSC FDC37M81x PC98/99 Compliant Enhanced Super I/O Controller with Keyboard/Mouse Wakeup Datasheet, 2000

# Parallel Port Registers

|                          |     |       |     |                         |      |                          |                            |               |
|--------------------------|-----|-------|-----|-------------------------|------|--------------------------|----------------------------|---------------|
| D7                       | D6  | D5    | D4  | D3                      | D2   | D1                       | D0                         | 0x378 Data    |
| $\overline{\text{Busy}}$ | Ack | Paper | Sel | Err                     |      |                          |                            | 0x379 Status  |
|                          |     |       |     | $\overline{\text{Sel}}$ | Init | $\overline{\text{Auto}}$ | $\overline{\text{Strobe}}$ | 0x37A Control |

1. Write Data
2. Assert Strobe
3. Wait for Busy to clear
4. Wait for Acknowledge



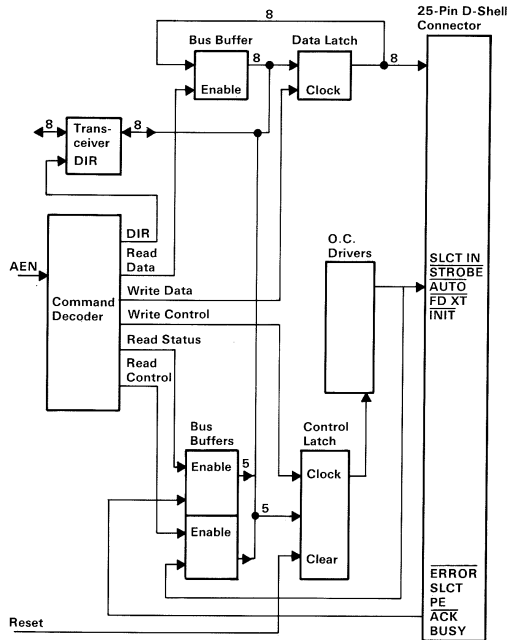
# A Parallel Port Driver

```
#define DATA    0x378
#define STATUS   0x379
#define CONTROL  0x37A

#define NBSY 0x80
#define NACK 0x40
#define OUT  0x20
#define SEL  0x10
#define NERR 0x08
#define STROBE 0x01

#define INVERT (NBSY | NACK |          SEL | NERR)
#define MASK   (NBSY | NACK | OUT | SEL | NERR)
#define NOT_READY(x) ((inb(x)^INVERT)&MASK)

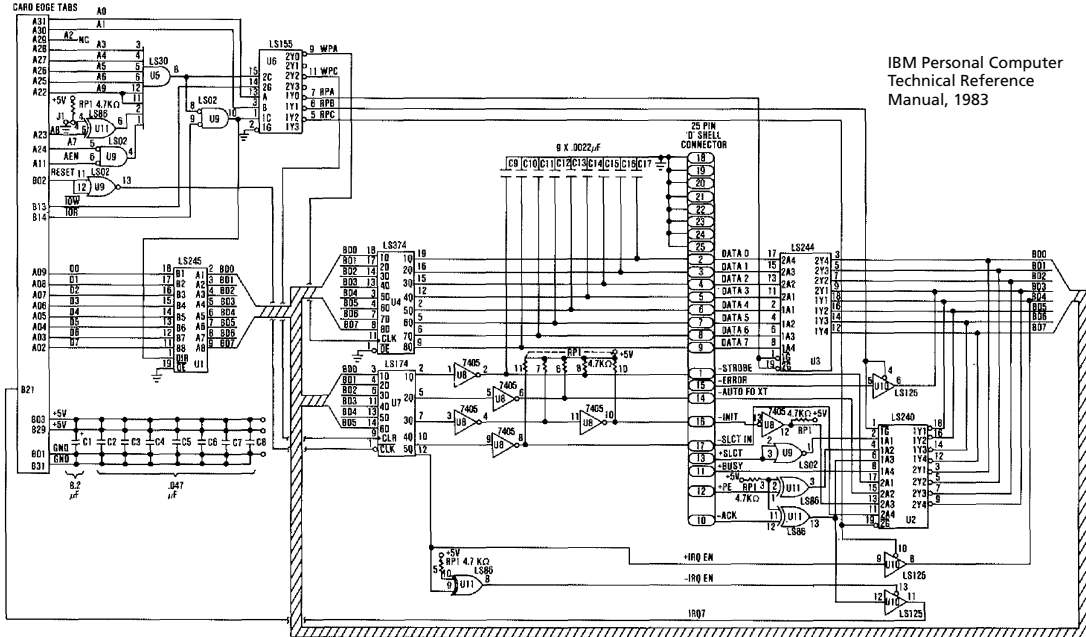
void write_single_character(char c) {
    while (NOT_READY(STATUS)) ;
    outb(DATA, c);
    outb(CONTROL, control | STROBE); /* Assert STROBE */
    outb(CONTROL, control ); /* Clear STROBE */
}
```



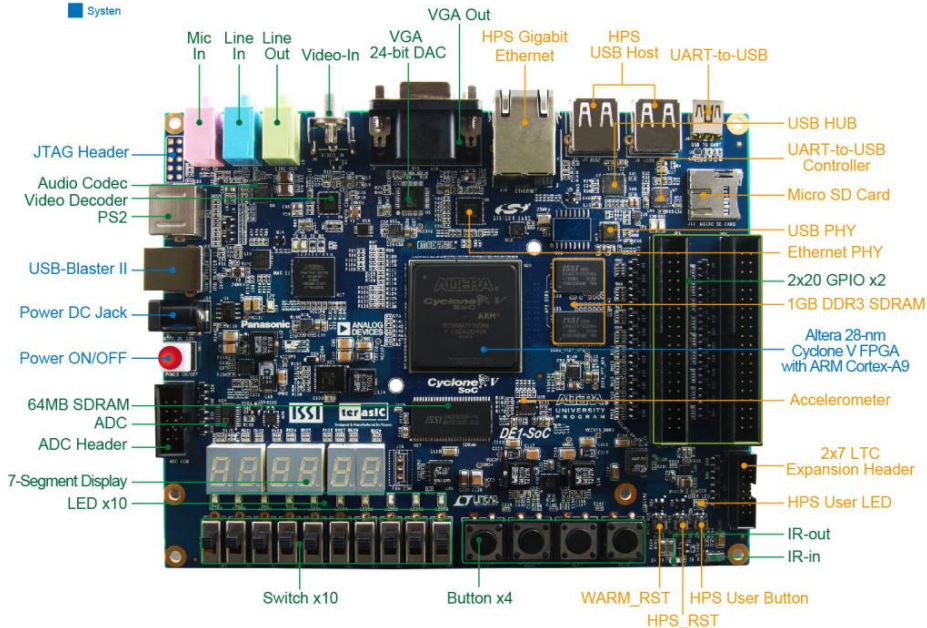
Printer Adapter Block Diagram

IBM Personal Computer  
Technical Reference  
Manual, 1983

## CARD EDGE TABS



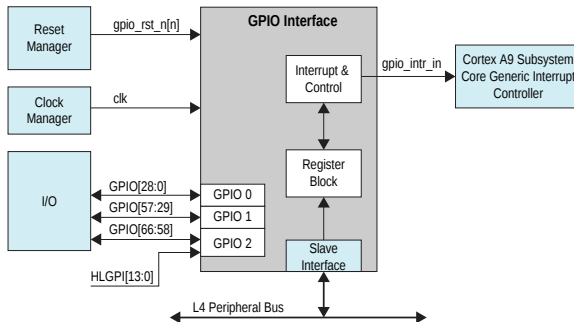
- FPGA
- HPS
- System



**Table 3-23 Pin Assignment of LEDs, Switches and Push-buttons**

| <b><i>Signal Name</i></b> | <b><i>HPS GPIO</i></b> | <b><i>Register/bit</i></b> | <b><i>Function</i></b> |
|---------------------------|------------------------|----------------------------|------------------------|
| <b>HPS_KEY</b>            | <b>GPIO54</b>          | <b>GPIO1[25]</b>           | <b>I/O</b>             |
| <b>HPS_LED</b>            | <b>GPIO53</b>          | <b>GPIO1[24]</b>           | <b>I/O</b>             |

**Figure 22-1: Cyclone V SoC GPIO**



Cyclone V Device Handbook  
Volume 3: Hard Processor  
System Technical Reference  
Manual  
Altera, 2013

**Table 22-1: GPIO Interface pin table**

| Pin Mux Name | Mapped to signal name | Comments       |
|--------------|-----------------------|----------------|
| GPIO [28:0]  | GPIO 0 [28:0]         | Input / Output |
| GPIO [57:29] | GPIO 1 [28:0]         | Input / Output |
| GPIO [66:58] | GPIO 2 [8:0]          | Input / Output |
| HLGPI [13:0] | GPIO 2 [26:13]        | Input only     |

## GPIO Module Address Map

Registers in the GPIO module

| Module Instance | Base Address |
|-----------------|--------------|
| gpio0           | 0xFF708000   |
| gpio1           | 0xFF709000   |
| gpio2           | 0xFF70A000   |

### GPIO Module

| Register                           | Offset | Width | Access | Reset Value | Description                    |
|------------------------------------|--------|-------|--------|-------------|--------------------------------|
| <a href="#">gpio_swporta_dr</a>    | 0x0    | 32    | RW     | 0x0         | Port A Data Register           |
| <a href="#">gpio_swporta_ddr</a>   | 0x4    | 32    | RW     | 0x0         | Port A Data Direction Register |
| <a href="#">gpio_inten</a>         | 0x30   | 32    | RW     | 0x0         | Interrupt Enable Register      |
| <a href="#">gpio_intmask</a>       | 0x34   | 32    | RW     | 0x0         | Interrupt Mask Register        |
| <a href="#">gpio_inttype_level</a> | 0x38   | 32    | RW     | 0x0         | Interrupt Level Register       |
| <a href="#">gpio_int_polarity</a>  | 0x3C   | 32    | RW     | 0x0         | Interrupt Polarity Register    |
| <a href="#">gpio_intstatus</a>     | 0x40   | 32    | R0     | 0x0         | Interrupt Status Register      |
| <a href="#">gpio_raw_intstatus</a> | 0x44   | 32    | R0     | 0x0         | Raw Interrupt Status Register  |
| <a href="#">gpio_debounce</a>      | 0x48   | 32    | RW     | 0x0         | Debounce Enable Register       |
| <a href="#">gpio_porta_eoi</a>     | 0x4C   | 32    | W0     | 0x0         | Clear Interrupt Register       |
| <a href="#">gpio_ext_porta</a>     | 0x50   | 32    | R0     | 0x0         | External Port A Register       |
| <a href="#">gpio_ls_sync</a>       | 0x60   | 32    | RW     | 0x0         | Synchronization Level Register |
| <a href="#">gpio_id_code</a>       | 0x64   | 32    | R0     | 0x0         | ID Code Register               |
| <a href="#">gpio_ver_id_code</a>   | 0x6C   | 32    | R0     | 0x3230382A  | GPIO Version Register          |
| <a href="#">gpio_config_reg2</a>   | 0x70   | 32    | R0     | 0x39CFC     | Configuration Register 2       |
| <a href="#">gpio_config_reg1</a>   | 0x74   | 32    | R0     | 0x1FF0F2    | Configuration Register 1       |

## gpio\_ver\_id\_code

GPIO Component Version

| Module Instance | Base Address | Register Address |
|-----------------|--------------|------------------|
| gpio0           | 0xFF708000   | 0xFF70806C       |
| gpio1           | 0xFF709000   | 0xFF70906C       |
| gpio2           | 0xFF70A000   | 0xFF70A06C       |

Offset: 0x6C

Access: R0

| Bit Fields       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 31               | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| gpio_ver_id_code |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| R0 0x3230382A    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 15               | 14 | 13 | 12 | 11 | 10 | 9  | 8  | 7  | 6  | 5  | 4  | 3  | 2  | 1  | 0  |
| gpio_ver_id_code |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| R0 0x3230382A    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

### gpio\_ver\_id\_code Fields

| Bit  | Name             | Description                                                                                                     | Access | Reset      |
|------|------------------|-----------------------------------------------------------------------------------------------------------------|--------|------------|
| 31:0 | gpio_ver_id_code | ASCII value for each number in the version, followed by *. For example. 32_30_31_2A represents the version 2.01 | R0     | 0x3230382A |

## gpio\_swporta\_dr

This GPIO Data register is used to output data on the GPIO signals. Check the GPIO chapter in the handbook for details on how GPIO2 is implemented.

| Module Instance | Base Address | Register Address |
|-----------------|--------------|------------------|
| gpio0           | 0xFF708000   | 0xFF708000       |
| gpio1           | 0xFF709000   | 0xFF709000       |
| gpio2           | 0xFF70A000   | 0xFF70A000       |

Offset: 0x0

Access: RW

**!** **Important:** To prevent indeterminate system behavior, reserved areas of memory must not be accessed by software or hardware. Any area of the memory map that is not explicitly defined as a register space or accessible memory is considered reserved.

| Bit Fields      |    |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |
|-----------------|----|----|----|-----------------|----|----|----|----|----|----|----|----|----|----|----|
| 31              | 30 | 29 | 28 | 27              | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved        |    |    |    | gpio_swporta_dr |    |    |    |    |    |    |    |    |    |    |    |
|                 |    |    |    | RW 0x0          |    |    |    |    |    |    |    |    |    |    |    |
| 15              | 14 | 13 | 12 | 11              | 10 | 9  | 8  | 7  | 6  | 5  | 4  | 3  | 2  | 1  | 0  |
| gpio_swporta_dr |    |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |
| RW 0x0          |    |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |

### gpio\_swporta\_dr Fields

| Bit  | Name            | Description                                                                                                                                                                                                                                                                                                           | Access | Reset |
|------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|-------|
| 28:0 | gpio_swporta_dr | Values written to this register are output on the I/O signals of the GPIO Data Register, if the corresponding data direction bits for GPIO Data Direction Field are set to Output mode. The value read back is equal to the last value written to this register. Note that only bits[26:0] are implemented for gpio2. | RW     | 0x0   |

Start the serial console before powering on the board

```
screen /dev/ttyUSB0 115200
```

Interrupt the boot process by quickly pressing a key

```
U-Boot SPL 2013.01.01 (Jan 12 2019 - 19:40:48)
```

```
BOARD : Altera SOCFPGA Cyclone V Board
```

```
...
```

```
Net:    mii0
```

```
Warning: failed to set MAC address
```

```
Hit any key to stop autoboot:  0
```

```
SOCFPGA_CYCLONE5 #
```

This is the U-Boot command line

```
SOCFPGA_CYCLONE5 # help
```

```
md      - memory display
```

```
mw      - memory write (fill)
```

## Read the GPIO Version Register

*gpio\_ver\_id\_code*

```
SOCFPGA_CYCLONE5 # md ff70906c 1  
ff70906c: 3230382a      *802
```

## Read the state of the HPS USER BUTTON

*gpio\_ext\_porta*

```
SOCFPGA_CYCLONE5 # md ff709050 1  
ff709050: 1df7ffff      ....
```

```
SOCFPGA_CYCLONE5 # md ff709050 1  
ff709050: 1ff7ffff      ....
```

## Set the data direction register to output for HPS USER LED

*gpio\_swporta\_ddr*

```
SOCFPGA_CYCLONE5 # mw ff709004 1000000
```

## Blink the LED

*gpio\_swporta\_dr*

```
SOCFPGA_CYCLONE5 # mw ff709000 1000000  
SOCFPGA_CYCLONE5 # mw ff709000 0
```

# Interrupts and Polling

Two ways to get data from a peripheral:

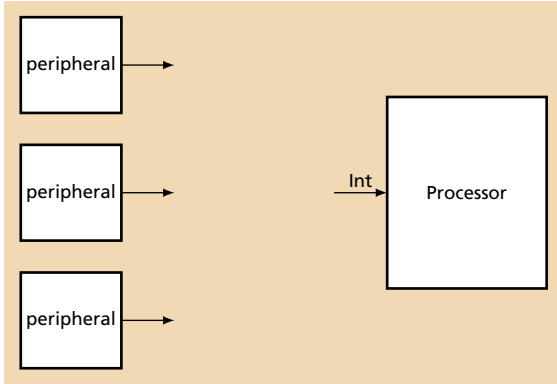
- ▶ Polling: "Are we there yet?"
- ▶ Interrupts: Ringing Telephone

# Interrupts

Basic idea:

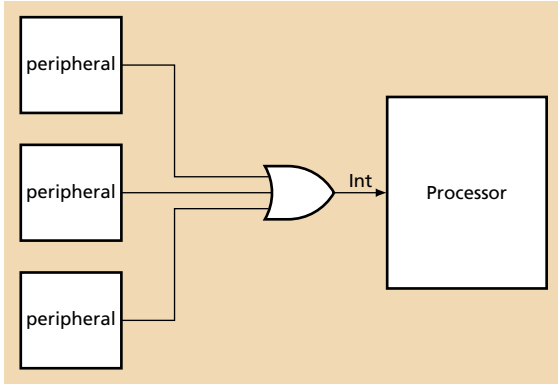
1. Peripheral asserts a processor's interrupt input
2. Processor temporarily transfers control to interrupt service routine
3. ISR gathers data from peripheral and acknowledges interrupt
4. ISR returns control to previously-executing program

# Many Different Interrupts



What's a processor to do?

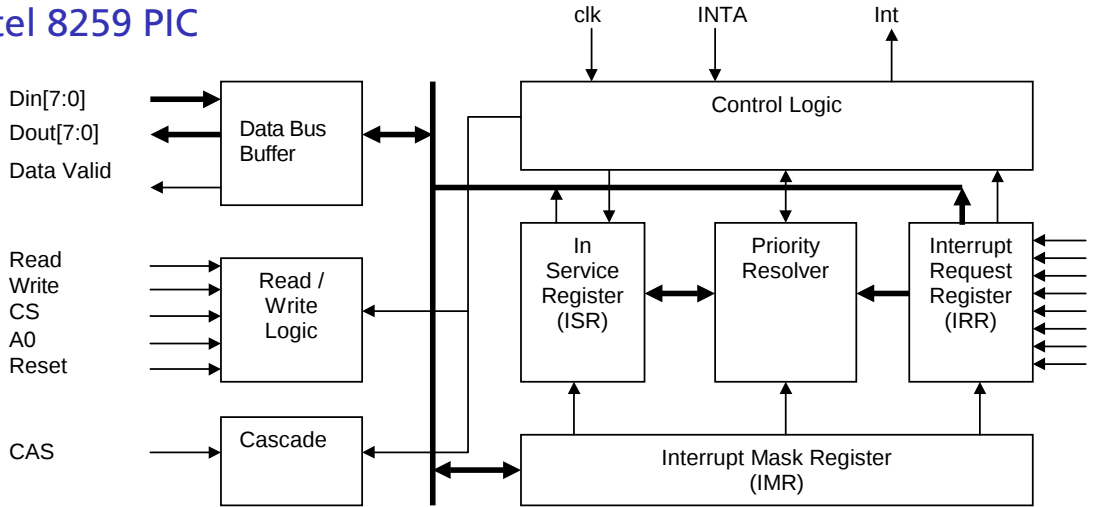
# Many Different Interrupts



What's a processor to do?

ISR polls all potential interrupt sources, then dispatches handler.

# Intel 8259 PIC



Prioritizes incoming requests & notifies processor

ISR reads 8-bit interrupt vector number of winner

IBM PC/AT: two 8259s; became standard

## Interrupts on the Cyclone V

Interrupt controller is part of the Cortex-A9 MPCore subsystem

The *Generic Interrupt Controller* supports 180 interrupt sources, including 64 from FPGA peripherals

Described in ARM's *Cortex-A9 MPCore Technical Reference Manual* (124 pages) and the *ARM Generic Interrupt Controller Architecture Specification* (952 pages)

Complicated by multiple cores, protection levels, and virtual machines

Many, many configuration registers. Each interrupt has a priority, enable flag, status, and control

A perfect thing to let the operating system manage:

<https://docs.kernel.org/core-api/genericirq.html>

# The NoCallerID Project, 2023

Asking the kernel to route interrupts from the device to our code:

```
// Get the interrupt number for our device from the Device Tree
int irq = irq_of_parse_and_map(pdev->dev.of_node, 0);
dev.irq_num = irq;

// Claim the interrupt and register our ISR
ret = request_irq(irq, (irq_handler_t) irq_handler, 0,
                  "csee4840_audio", NULL);
```

Code the kernel will call when the interrupt occurs:

```
// Our Interrupt Service Routine, called by the kernel
irq_handler_t irq_handler(int irq, void *dev_id,
                          struct pt_regs *reg) {
    ...
    wake_up_interruptible(&wq); // Unblock the program waiting on us

    return IRQ_RETVAL(1);
}
```