

# CPU TIMING and SYSTEM BUS INTERFACING

## EE3232 DIGITAL SYSTEMS III CLASS NOTES CHAPTER 13

Department of Electrical Engineering  
University of New Brunswick

© C.P. Diduch

December 5, 1999

### SUMMARY

- Objectives.
- CPU clocks.
- CPU reset.
- Reset hardware.
- Power on reset timing.
- Logic representation as a range of voltages.
- Gate loading for bipolar semiconductors.
- Bipolar semiconductor loading models.
- Fanout.
- Gate loading for MOS semiconductors.
- The PC104 bus.
- Memory interfacing.
- Memory device interfacing steps.
- ROM device timing.
- CPU device timing.
- CPU address access time.
- CPU read time.
- Accounting for buffer delay in ROM design.

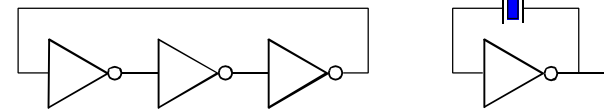
### OBJECTIVES

- To concisely explain:
  - ▶ Specifications on CPU clocks,
  - ▶ Need for power-on reset,
  - ▶ Wait state generation.

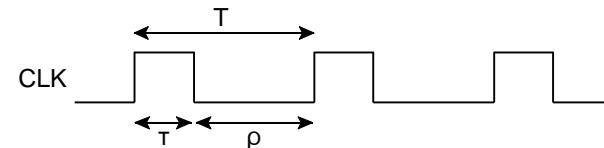
- To learn how to design power on reset circuits.
- From the gate specs :  $V_{IL}$ ,  $V_{IH}$ ,  $V_{OL}$ ,  $V_{OH}$ ,  $I_{IL}$ ,  $I_{IH}$ ,  $I_{OL}$  and  $I_{OH}$ , determine if a system of logic devices is guaranteed to function properly when interconnected.
- Given device timing specs determine the minimum number of wait states required to guarantee proper operation.

### CPU CLOCKS

- CPU operation is synchronized with the clock signal.
- To build a clock generator : use feedback around an odd number of inverters.



- The frequency of the clock depends on the delay (phase shift) around the feedback loop.
- The crystal reduces frequency variations caused by temperature and device switching delay.
- The specifications on microprocessor clocks include,
  - ▶  $\tau$ , the clock hi-time,
  - ▶  $\rho$ , the clock lo-time,
  - ▶  $T$ , the clock period and
  - ▶  $f$ , the clock frequency,  $f = T^{-1}$ ,  $T = \tau + \rho$ .



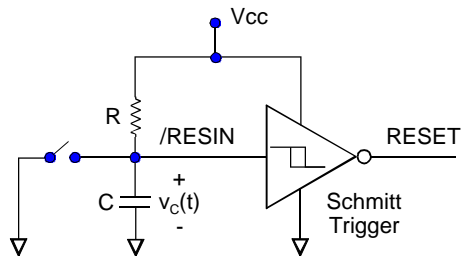
- $\tau$  and  $\rho$  are specified as minima.
- A usual specification :  $f_{min} < f < f_{max}$  **or**  $f < f_{max}$ .
  - ▶  $f_{min}$  : is the limit of operation for dynamic RAM.
  - ▶  $f_{max}$  : is limited by gate delay and power dissipation.

## CPU RESET

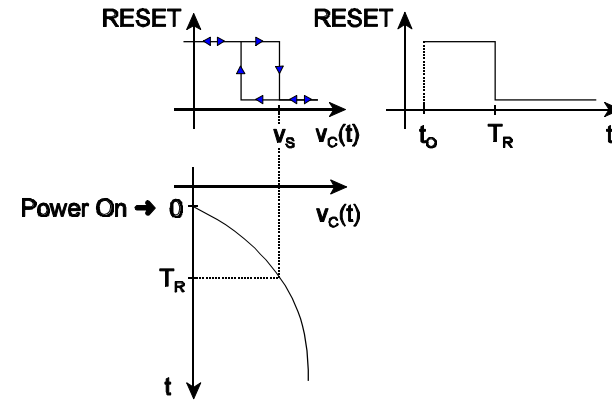
- Activating /RESIN forces the CPU to begin fetching instructions from the **reset address**.
- Reset in the 80C188XL causes:
  - (FLAGS) ← (IP) ← (DS) ← (ES) ← 0000H
  - (CS) ← FFFFH
  - QUEUE ← emptied
- The reset address for the 80C188XL is ∴ FFFF0H.
- Instructions at the reset address must reside in non-volatile memory, i.e., ROM or EPROM ...
- The reset signal must be active for a minimum specified time,  $T_R$ , to guarantee a proper reset.

## RESET H/W

- Hardware for implementing a power-on / manual reset,
  - An RC network,
  - Manual switch and
  - Schmitt trigger (to **sharpen** the /RESIN signal).



## POWER ON RESET TIMING



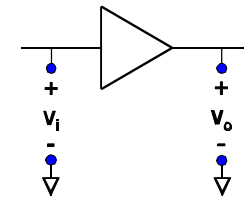
$$v_c(t) = v_{CC} \left( 1 - e^{-\frac{t}{RC}} \right)$$

$$v_c(T_R) = v_{CC} \left( 1 - e^{-\frac{T_R}{RC}} \right) = v_s$$

$$T_R = -RC \ln \left( 1 - \frac{v_s}{v_{CC}} \right)$$

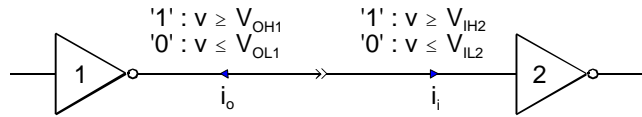
## LOGIC REPRESENTATION AS VOLTAGE RANGES

- Logic '0' and logic '1' are represented by **voltage ranges** in many electronic devices.
- Denote input gate voltage,  $v_i$ , and output gate voltage,  $v_o$ .



- Logic '0' at the gate output is represented by  $v_o \leq V_{OL}$ .
- Logic '1' at the gate output is represented by  $v_o \geq V_{OH}$ .
- Logic '0' at the gate input is represented by  $v_i \leq V_{IL}$ .
- Logic '1' at the gate input is represented by  $v_i \geq V_{IH}$ .

## LOGIC SIGNAL REPRESENTATION (CONT'D)



- If Gate<sub>1</sub> drives Gate<sub>2</sub> then the following conditions must be satisfied to ensure proper logical operation,

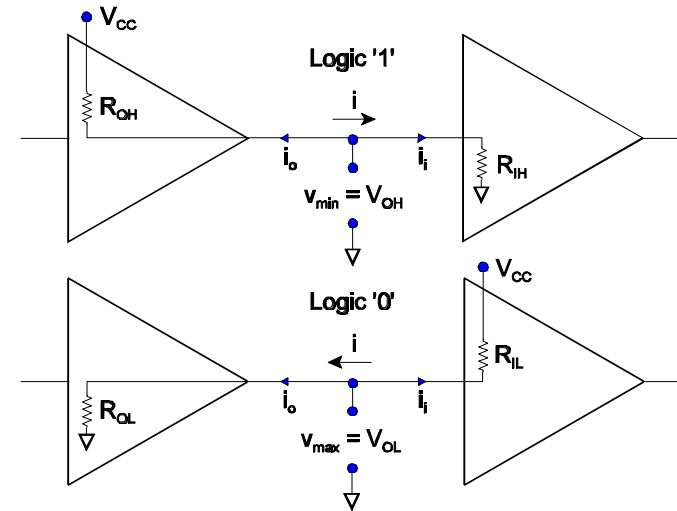
$$V_{OL1} \leq V_{IL2} \quad \text{and} \quad V_{OH1} \geq V_{IH2}$$

- Under these conditions the gates are said to possess a **Noise Margin**,
  - Logic Lo Noise Margin =  $V_{IL2} - V_{OL1}$ .
  - Logic Hi Noise Margin =  $V_{OH1} - V_{IH2}$ .

## GATE LOADING FOR BIPOLAR SEMICONDUCTORS

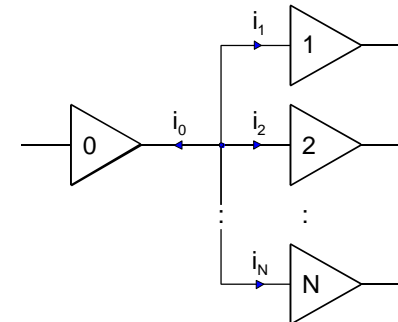
- Bipolar semiconductor loading is predominantly resistive.
- There are upper limits on the **output current** that a gate can source or sink at logic '1' and '0'.
- There are also specs on the maximum **input current** that a gate will source or sink when the input is driven at logic '1' and '0'.
  - If the input = '0',  $|i_i| < |I_{IL}|$  (guaranteed maximum).
  - If the input = '1',  $|i_i| < |I_{IH}|$  (guaranteed maximum).
  - If the output = '0',  $|i_o| < |I_{OL}|$  (guaranteed maximum).
  - If the output = '1',  $|i_o| < |I_{OH}|$  (guaranteed maximum).
- NOTE**: it is a convention that **positive** current directions are always **into** the gate.

## BIPOLAR SEMICONDUCTOR LOADING MODELS



## BIPOLAR SEMICONDUCTOR LOADING - FANOUT

- There are limits on the number of gates that may be driven by a single gate output.
- For the following assume  $I_{OH} < 0$ ,  $I_{IH} > 0$ ,  $I_{OL} > 0$ ,  $I_{IL} < 0$ .



## FANOUT (CONT'D)

- At Logic '1' :

$$-i_0 = i_1 + i_2 + \dots + i_N$$

$$i_1 \leq I_{IH1}, \quad i_2 \leq I_{IH2}, \quad i_N \leq I_{IHN}$$

- To guarantee proper gate operation (at Logic '1') :

$$|I_{OH0}| \geq |I_{IH1}| + |I_{IH2}| + \dots + |I_{IHN}|$$

- At Logic '0' :

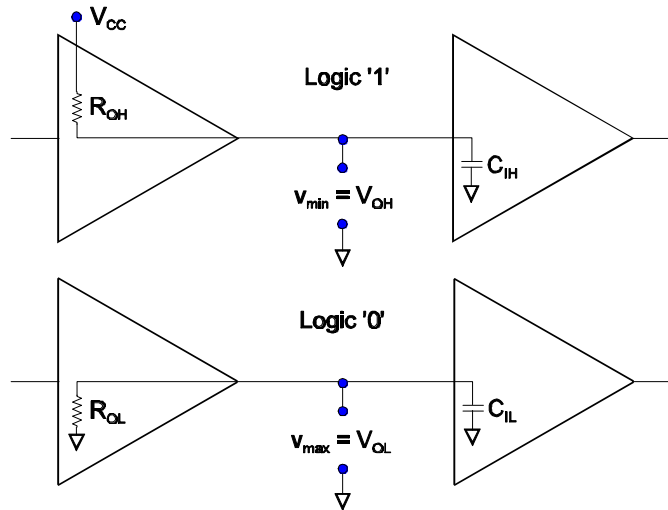
$$i_0 = -(i_1 + i_2 + \dots + i_N)$$

$$-i_1 \leq -I_{IL1}, \quad -i_2 \leq -I_{IL2}, \quad -i_N \leq -I_{ILN}$$

- To guarantee proper gate operation (at Logic '0') :

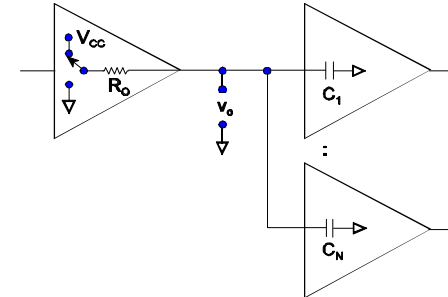
$$|I_{OL0}| \geq |I_{IL1}| + |I_{IL2}| + \dots + |I_{ILN}|$$

## GATE LOADING FOR MOS SEMICONDUCTORS



## GATE LOADING FOR MOS SEMICONDUCTORS (CONT'D)

- MOS semiconductor loading is predominantly capacitive.
- Capacitive loading increases the propagation delay.
- As additional MOS gates are connected to an output pin, the equivalent  $R_o C_{EQU}$  time constant increases.



- As additional MOS gates are connected to an output pin the effective signal delay increases.
- Limits exist on the number of gates that may be driven by a single gate output → limits depend on the maximum acceptable delay.
- Sketch the gate OP voltage,  $v_o$ , for changes in output logic as the fanout,  $N$ , increases.

## THE PC104 BUS

- The PC104 bus standard specifies the physical and electrical aspects of an 8-bit microcomputer system bus. The bus is organized into four groups,
  - Power Bus,
  - Data Bus,
  - Address Bus and
  - Control Bus.

## MEMORY INTERFACING

---

- Given the required memory **capacity**, design a system bus interface using specified memory devices.
- If the system address bus is 20 bits then the maximum memory capacity is,
- $2^{20}$  Bytes =  $2^{10} \times 2^{10}$  Bytes = 1024K Bytes = 1M Byte
- $1K = 2^{10}$  ,  $1M = 2^{20}$
- A **memory map** shows the type of memory devices that occupy an address space and the associated starting and ending addresses.

## MEMORY DEVICE INTERFACING STEPS

---

- Draw a memory map for the interface.
- Connect the **lower** system bus address lines,  $a_0 - a_{K-1}$ , to the K address pins of each memory device.
- Generate the chip selects for each memory device by decoding the **upper** address lines,  $a_K - a_m$ , consistent with the address map.
- Interface the data lines, the /MEMRD and where appropriate, the /MEMWR control lines.
- Determine the requirement for system bus drivers and transceivers.
- Determine the timing requirements and need for wait states.

## ROM DEVICE TIMING

---

- A ROM device may be modeled as follows.
  - A time delay exists between the instant when an address or enable is applied to a ROM device and the instant when valid data appears at the output pins.
- $t_{ACC}$ : **Memory Access Time**: maximum specification on  $\tau_1$ , assuming /CE = /OE = active.
- $t_{CE}$ : **Chip Enable to Output Time**: maximum specification on  $\tau_2$ , assuming /OE = active and a valid address.

$t_{OE}$ : **Output Enable to Output Time**: maximum specification on  $\tau_3$ , assuming /CE = active and a valid address.

## CPU DEVICE TIMING

---

- A simplified representation of read cycle timing specifications for an 80C188XL appears below,

## CPU ADDRESS ACCESS TIME

---

- CPU **address access time**: the time between generation of a valid address on the address pins of the CPU and the instant data is read by the CPU.
- For the 80C188XL microprocessor, the **CPU address access time** is given by,

$$(T_1 - T_{CLAV}) + T_2 + (T_3 - T_{DVCL}) + nT_w$$

- where,
  - $T_i$  = duration of one T-state,
  - $n$  = number of wait states inserted in a bus cycle,
  - $T_w$  = duration of one wait state,
  - $T_{CLAV}$  = duration between the start of  $T_1$  and a valid address,
  - $T_{DVCL}$  = read data setup, the time that data must be valid before the end of  $T_3$ .

## CPU READ TIME

---

- CPU **read time** is defined as the time between the generation of an active read control signal and the instant data is read by the CPU.
- For the 80188, the **CPU read time** is given by,

$$(T_2 - T_{CLRL}) + (T_3 - T_{DVCL}) + nT_w$$

$T_{CLRL}$ : Read delay, the duration between the end of  $T_2$  and active /RD.

## ACCOUNTING FOR BUFFER DELAY IN ROM DESIGN

---

- To guarantee proper system operation,
  1. **CPU address access time**  $> t_{ACC} + T_{AD}$
  2. **CPU address access time**  $> t_{CE} + T_{DE}$
  3. **CPU read time**  $> t_{OE} + T_{CD}$
- $T_{AD} = T_1 + T_2 + T_4 + T_5$  : represents the total worst case delay caused by devices connected between the address and data pins of the CPU and ROM.
- $T_{DE} = T_1 + T_2 + T_3 + T_4 + T_5$  : represents the total worst case delay caused by devices connected between the CPU address pins, the /CE pin of the ROM and the data pins.
- $T_{CD} = T_4 + T_5 + T_6 + T_7$  : represents the total worst case delay caused by devices connected between the CPU /RD pin and data pins and the ROM /OE and data pins.