Parallel Buses

ECE 153B

Sensor & Peripheral Interface Design

Winter 2016

Sensor & Peripheral Interface Design

- Keyword is Interface
  - In this instance, it refers to the interface between a processor and its peripherals in a digital system
- Sensors and peripherals (I/O devices), as well as memory, normally interface to a processor via buses
- Definition of a bus
  - From the Latin omnibus meaning “for all”
  - A shared interconnect within a digital system
    - As opposed to dedicated, point to point connections
  - Sharing managed through an agreed upon bus protocol
- If a device can communicate with the bus (using the bus protocol) then it can be used in a digital system
  - Bus connected devices “speak” the same language

Bus Structure – Isolated I/O

- Early bus architectures separated memory access from I/O access
  - Memory access tightly coupled to processor
    - Instruction and data fetch from magnetic core memory
  - I/O access handled via a “channel” (or channel controller)
    - A channel is essentially a small computer dedicated to the handling of I/O with peripherals
    - Channels allowed the efficient use of interrupts generated by peripherals
    - Channels were introduced on the IBM 709 in 1958
  - This structure requires separate I/O instructions
    - Early Intel microprocessors had IN and OUT instructions (for peripherals) and MOV instructions for memory and register transfers
    - Advantage is that it saves limited memory address space
      - 16 bit addresses = 64K of memory

Bus Structure – Memory Mapped I/O

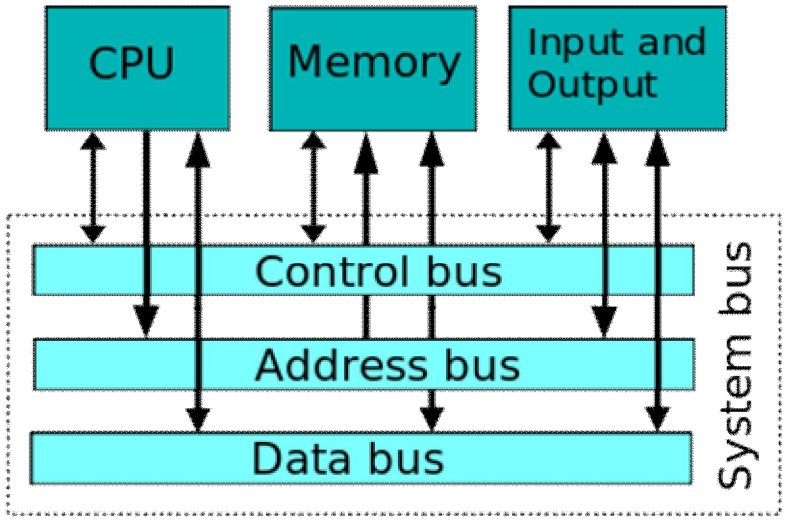
- Later, mini and micro computers made use of memory mapped I/O
  - Maps peripherals onto the memory bus so that the input and output devices appear to be memory locations
    - This was first implemented on the Digital Equipment Corp. (DEC) Unibus of the PDP-11 around 1969
  - Advantages:
    - No special I/O-related instructions or datapaths needed
    - Simplifies and standardizes processor design
  - Disadvantages:
    - Device interfaces must decode their addresses and adapt to processor's bus cycles .... added complexity
    - Memory and I/O devices utilize same clock…difficult to increase clock frequency

Bus Components

- Data Bus
  - Typical data widths of parallel buses are 8, 16, 32, 64
  - Serial buses exist too (serial ⇒ width = 1)
    - Much more on serial buses later
- Address Bus
  - Often, but not always, same as data bus wires (shared or multiplexed)
    - Separate address and data buses on ECE 153B processor
- Control Signals (or Control Bus)
  - Clock (or clocks)
  - Bus arbitration signals (Request & Acknowledge)
  - Direction control (read vs. write)
  - I/O vs. memory access
  - Internal vs. external access
  - Width specifiers (byte, word, long word, etc.)
  - Parity and error control/status signals
  - Cycle framers and timing edges
  - Interrupts

System Bus Structure

- General System Bus Structure (below) applies to both isolated and memory mapped I/O
  - Address and Data buses are normally shared in both structures
  - Differences are handled via Control bus



Bus Interface Design

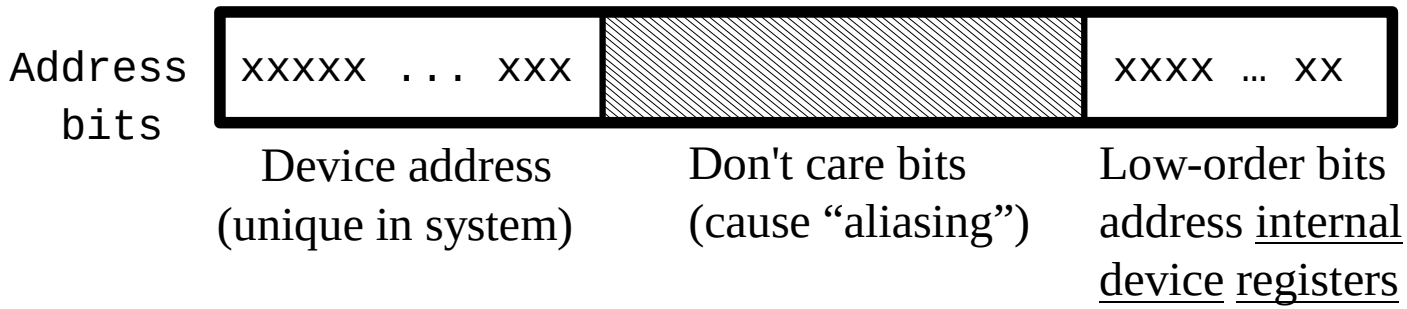
- Bus interface design involves generating the control signals the peripheral requires from the control signals the processor generates
- In a “chip set”, both processor and peripheral come from the same vendor (or second source) and the connection is direct
- When interfacing a processor from one vendor with a peripheral from another vendor, a hardware interface must be designed
  - Generally accomplished via programmable logic (i.e., CPLD, FPGA, etc.)
    - Bus control CPLD (U101) on E153BSYS board

Bus Interface Design

- Bus Interface Design involves satisfying the following four requirements
  - Device Address Decoding
    - Recognize a unique address (or address range) and ensure that it is only addressed during valid cycles and has correct transfer direction
  - Sequence Domain Constraints
    - Determine the order and relationship of events in a bus transaction
  - Time Domain Constraints
    - Determine if device meets timing requirements
      - Add wait states or select different device
  - Electrical Requirements
    - Ensure that logic levels and voltage, current and power requirements are satisfied

Device Address Decoding

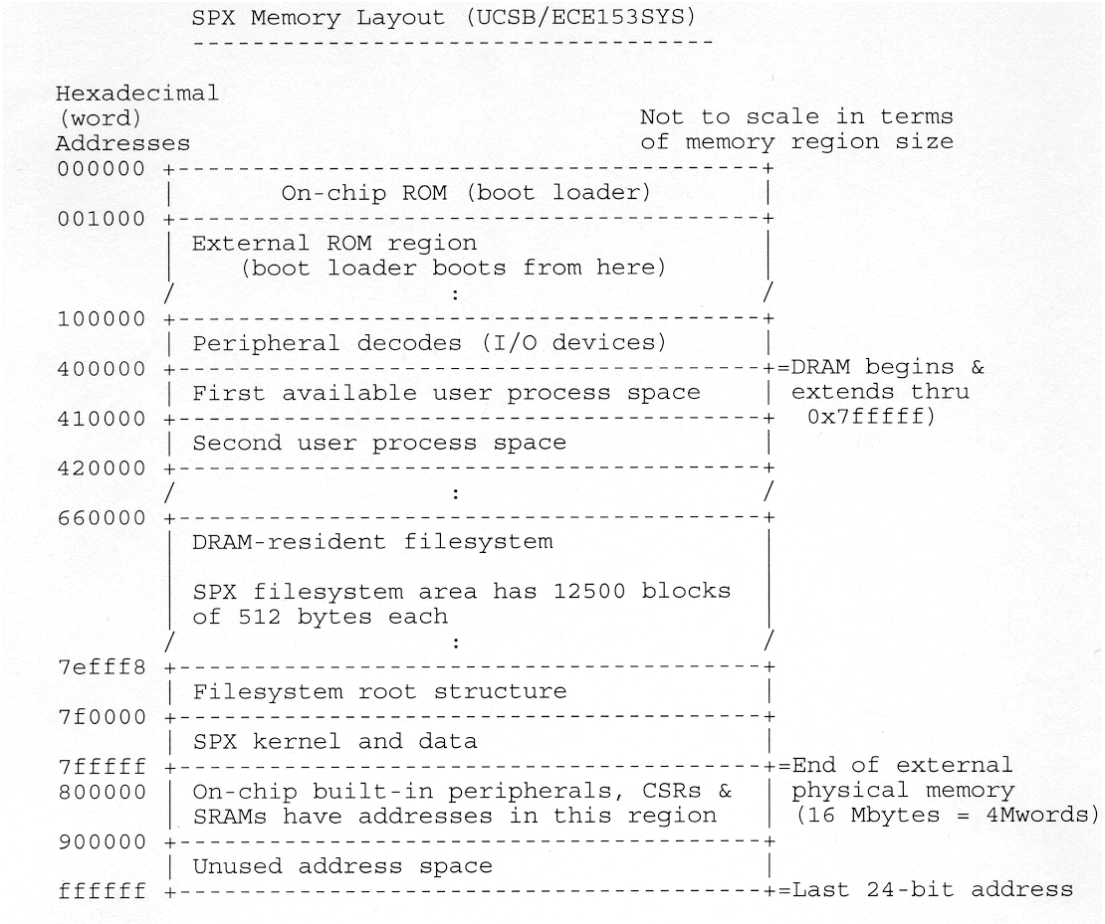
- Memory mapped I/O technique maps a *range* of regular address space onto a given device
  - Any processor access to this range goes to the device
  - Device sees address (i.e. its "name") and responds as a slave in master/slave bus cycle



Device Address Decoding

- Implementation requires an address decoder
  - Device address derived from high-order bus address bits
  - Matches with device's unique address pattern during active bus cycles
  - Low-order bits used to address internal registers
  - Generate appropriate READ, WRITE and RESET control signals from processor control signals

ECE153BSYS Memory Map



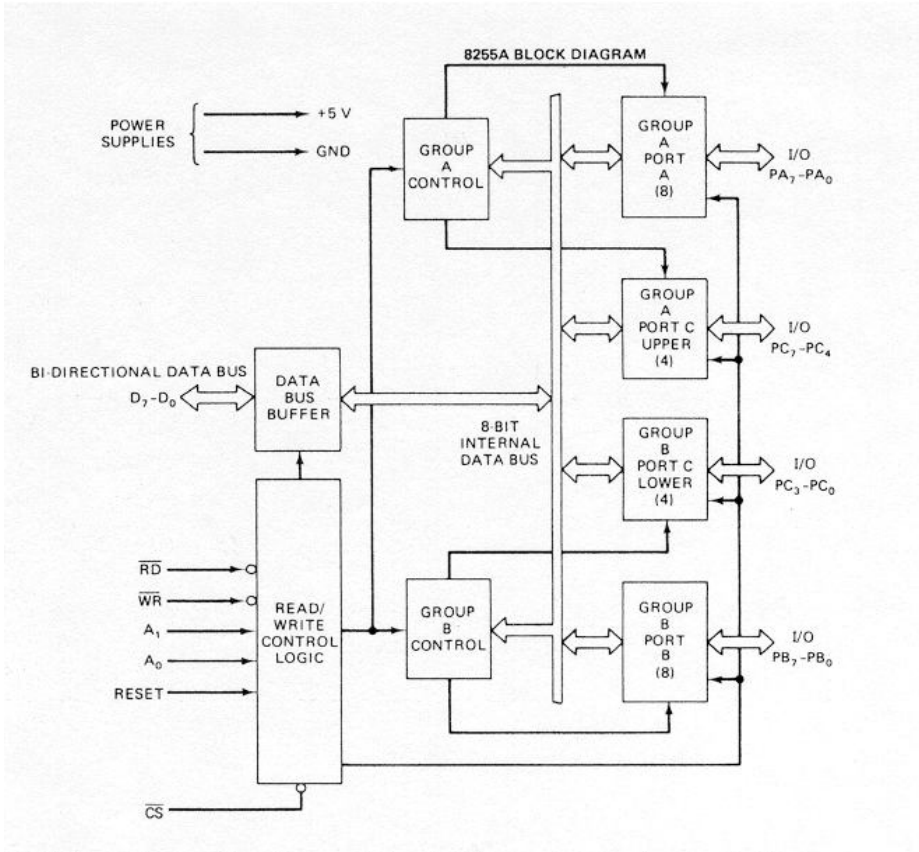
Address Decoder Design Example

- The table below illustrates the data, address and (portions of) the control buses for the Texas Instruments TMS32C031 DSP (the processor on the ECE 153B SYS board)
  - 32 bit data bus
  - 24 bit address bus
  - Single R/W* direction control signal
  - STRB* signal indicating external access
  - RESET * signal

TERMINAL NAME	TERMINAL QTY	TYPE†	DESCRIPTION
			PRIMARY-BUS INTERFACE
D31–D0	32	I/O/Z	32-bit data port
A23–A0	24	O/Z	24-bit address port
R/$\overline{W}$	1	O/Z	Read/write. R/$\overline{W}$ is high when a read is performed and low when a write is performed over the parallel interface.
$\overline{STRB}$	1	O/Z	External-access strobe
			CONTROL SIGNALS
$\overline{RESET}$	1	I	Reset. When $\overline{RESET}$ is a logic low, the device is in the reset condition. When $\overline{RESET}$ becomes a logic high, execution begins from the location specified by the reset vector.

Address Decoder Design Example

- The Intel 8255A Programmable Peripheral Interface
  - Three addressable 8 bit ports
  - 8 bit data bus
  - Discrete RD* and WR* signals
  - High true RESET signal
  - Low true CS* (chip select signal)



Address Decoder Design Example

- 8255A Basic Operation

TABLE 8.6-1

8255A BASIC OPERATION [3]

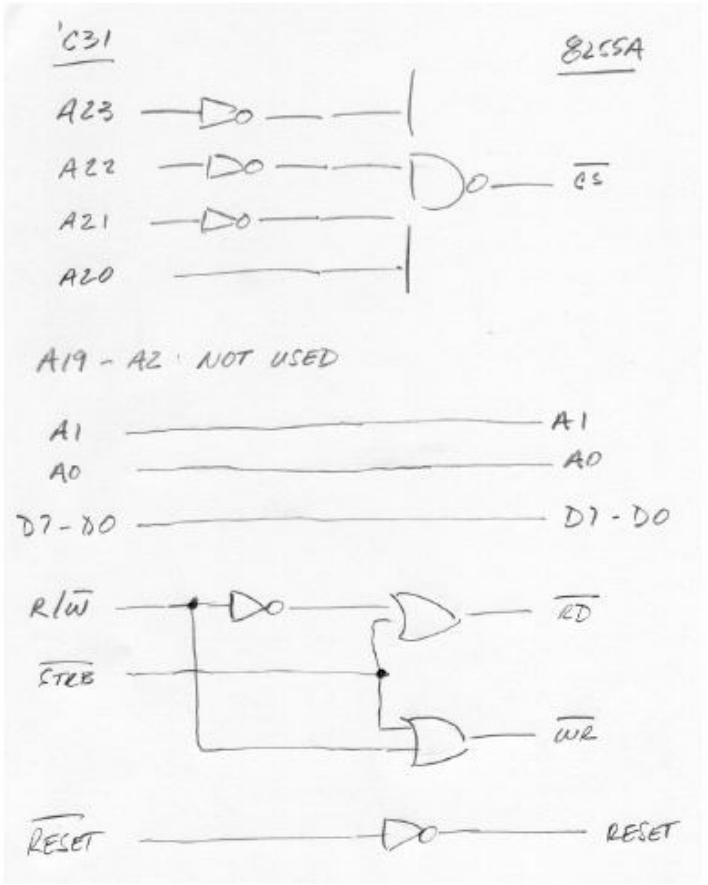
A_1	A_0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	INPUT OPERATION (READ)
0	0	0	1	0	PORT A ⇒ DATA BUS
0	1	0	1	0	PORT B ⇒ DATA BUS
1	0	0	1	0	PORT C ⇒ DATA BUS
					OUTPUT OPERATION (WRITE)
0	0	1	0	0	DATA BUS ⇒ PORT A
0	1	1	0	0	DATA BUS ⇒ PORT B
1	0	1	0	0	DATA BUS ⇒ PORT C
1	1	1	0	0	DATA BUS ⇒ CONTROL
					DISABLE FUNCTION
X	X	X	X	1	DATA BUS ⇒ 3-STATE
1	1	0	1	0	ILLEGAL CONDITION
X	X	1	1	0	DATA BUS ⇒ 3-STATE

Address Decoder Design Example

- Assuming that this is the only peripheral device connected to the ‘C31 processor, design the interface circuitry to place it at the lowest address(es) in the peripheral decode space
- Memory map shows peripherals located from address (hex) 10 0000 to 3F FFFF ($(0001)_b0\ 0000_h$) to ($(0011)_bF\ FFFF_h$)
  - 8255A should be at address $(0001)_b0\ 0000_h$
  - Internal ports and registers should be at:
    - Port A: 1X XXX xx00
    - Port B: 1X XXX xx01
    - Port C: 1X XXX xx10
    - Control: 1X XXX xx11

Address Decoder Design Example

- Four most significant bits of address bus used for chip select (CS*)
  - 1X XXXX enables chip
- Two least significant bits used to address ports A, B and C and control register
- 8255A data I/O connected to 8 least significant bits of DSP data bus
- Individual RD* and WR* signals derived from DSP R/W* and STRB* (external access) signals
- Polarity of 8255A RESET signal is high true vs. low true RESET* signal from DSP
- Decoder easily realizable using programmable logic device (PLD, FPGA, etc.)



Address Decoder Design Example

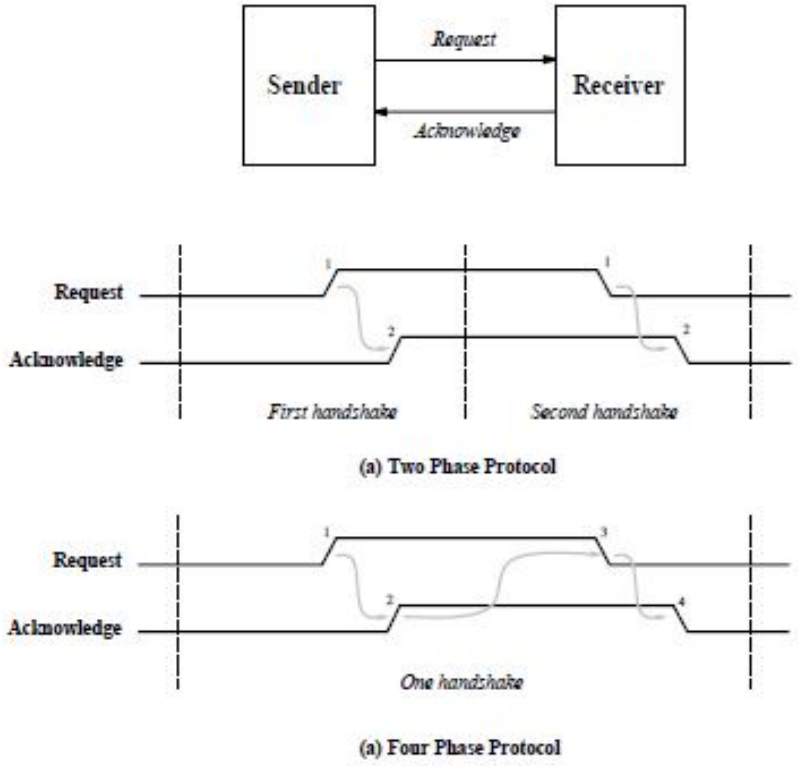
- Address of 8255A is not fully decoded
  - To fully decode address, all 23 bits of address bus would be utilized
    - A23 – A2 = 10 0000 for chip select (CS*)
    - A1 – A0 = Port / Control select
- Not necessary in this design because the 8255A is the only peripheral and the memory map defines the peripheral space as 10 0000 through 3F FFFF
- This results in “aliasing” since any address between 10 0000 and 1F FFFF will select this device
  - This approach requires much less hardware than fully decoding the address
  - If additional peripherals are included in the design, progressively more address bits can be included
  - The only requirement is that the peripheral addresses are mutually exclusive and all reside in the peripheral space of the memory map

Sequence Domain Constraints

- Once the address decoding of the device has been determined, the next step in designing the bus interface is satisfying the sequence domain constraints
- At a high level, this requires that you:
  - Thoroughly understand the cycles that will be seen on the bus side
  - Thoroughly understand the operation of the device being interfaced
  - Plan your own device transactions so that they will be as compatible as possible with bus cycles
- Formalizing the above description, we assert that in the sequence domain, events are ordered such that
  - A precedes B, or
  - A and B are concurrent (no relationship), or
  - A and B are identical

Sequence Domain Constraints

- Two Phase (Cycle) & Four Phase (Cycle), Equipotential (Isochronic) Signaling



Sequence Domain Constraints

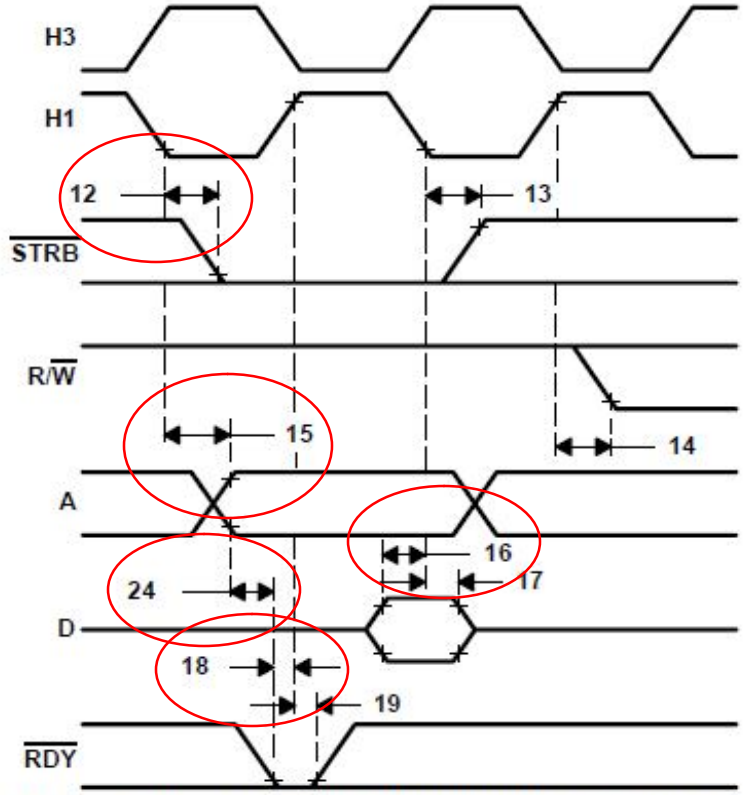
- If the timing information is stripped out of the diagram on the following page, we are left with the sequence domain constraints
- Three additional signals (H1, H3 and RDY*) are required to effect handshaking and define sequence domain operation

SUPPLY AND OSCILLATOR SIGNALS			
H1	1	O/Z	External H1 clock. H1 has a period equal to twice CLKIN.
H3	1	O/Z	External H3 clock. H3 has a period equal to twice CLKIN.

PRIMARY-BUS INTERFACE			
D31–D0	32	I/O/Z	32-bit data port
A23–A0	24	O/Z	24-bit address port
R/$\overline{W}$	1	O/Z	Read/write. R/$\overline{W}$ is high when a read is performed and low when a write is performed over the parallel interface.
$\overline{STRB}$	1	O/Z	External-access strobe
$\overline{RDY}$	1	I	Ready. $\overline{RDY}$ indicates that the external device is prepared for a transaction completion.

Sequence Domain Definition of TMS32C031 Read Operation

- H1 going low precedes STRB* going low (12)
- H1 going low precedes ADDRESS becomes valid (15)
- STRB* going low is concurrent with ADDRESS becoming valid (**)
- ADRRESS becoming valid precedes RDY* going low (24)
- RDY* going low precedes processor READ operation enabled (18)
  - RDY* is sampled and READ is enabled on rising edge of H1
- DATA becomes valid precedes H1 going low (16)
- RDY* going low is concurrent with DATA becoming valid (**)



NOTE A: $\overline{\text{STRB}}$ remains low during back-to-back read operations.

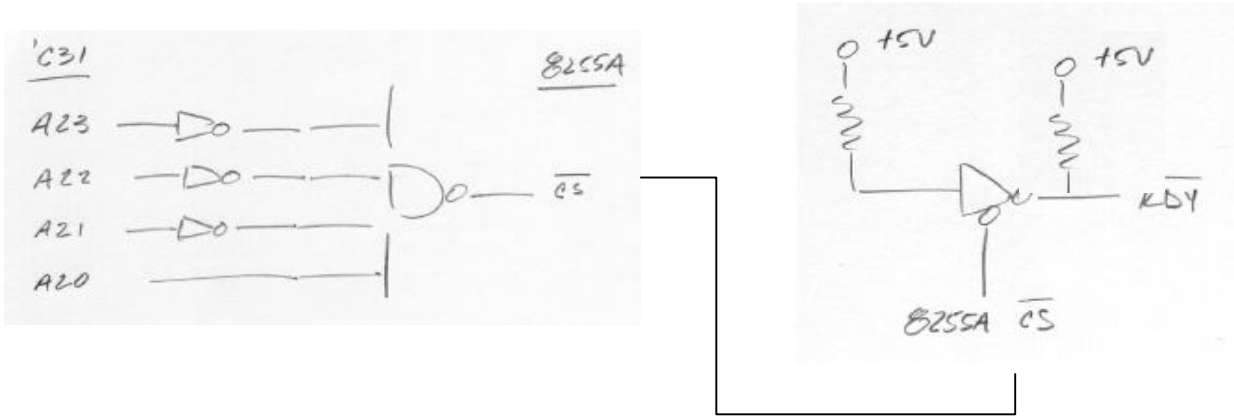
Incorporating Sequence Domain Constraints into Address Decoder Design Example

- The following sequence domain constraints are guaranteed by the processor
  - H1 going low precedes STRB* going low (12)
  - H1 going low precedes ADDRESS becomes valid (15)
  - STRB* going low is concurrent with ADDRESS becoming valid (**)
- The sequence domain constraint that must me satisfied by the interface is
  - ADRRESS becoming valid precedes RDY* going low (24)
- The sequence (and time) domain requirement that must be satisfied by the peripheral
  - DATA becomes valid precedes H1 going low (16)

Incorporating Sequence Domain Constraints into Address Decoder Design Example

- From Address Decoder Design Example,
  8255A CS* = (A23' • A22' • A21' • A20)'
  - i.e., Valid Address = 1XX XXXX
- Actual timing of Address bus vs. 8255A CS* vs. RDY* is unknown (at this point), but sequence domain characteristic of:

  "ADRRESS becoming valid precedes RDY* going low" is guaranteed



Time Domain Constraints

- The next step is adding time domain information to the sequence domain constraints
- The fundamental timing equation of a synchronous digital system:

$$\text{Clock period} \geq \text{CLK to Q} + \text{Prop Delay} + \text{Setup}$$

- The table on the following page attaches timing parameters to the waveform diagram introduced earlier (slide 21)
- The Delay times are simply the combinational min and max case times between event occurrences, generally signal edges to other signal edges or signal edges to data validity
  - Clk to Q and Prop Delay are delay times

Time Domain Constraints

- Setup and Hold times define the timing relationship between event occurrences (again, generally signal edges) and data validity for correct sequential operation. Specifically:
  - Setup time (t_{su})
    - the amount of time the data input to a flip-flop must be stable before the active signal edge in order to guarantee correct latching operation
  - Hold time (t_{hold})
    - the amount of time the data input to a flip-flop must remain stable after the active signal edge in order to guarantee correct latching operation

Time Domain Definition of TMS32C031-50 Read Operation

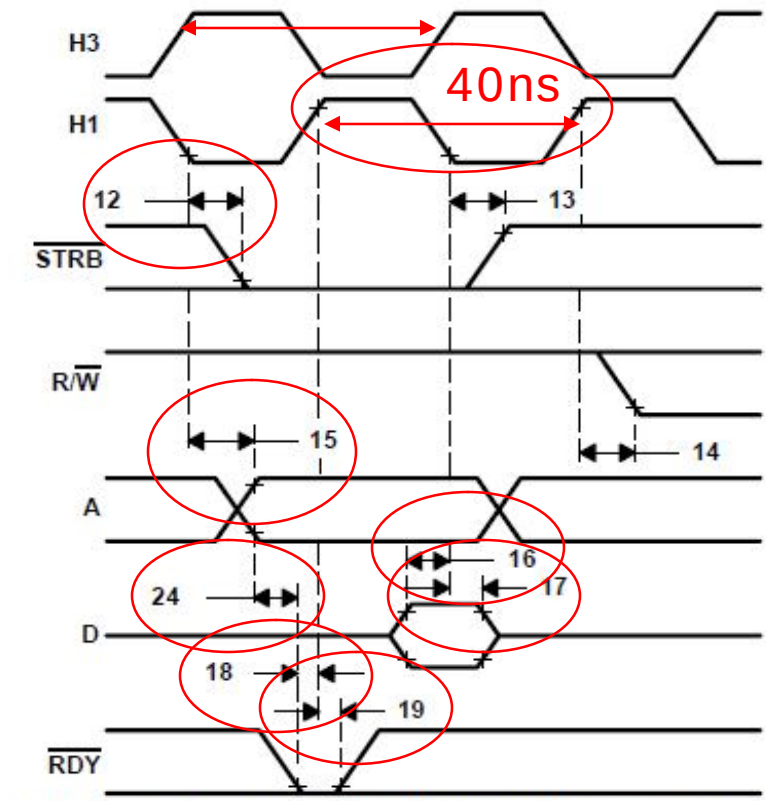
- The table below indicates the timing parameters for the various speed and voltage grades of the TI TMS30C31 processor
  - 'C31 indicates a 5V part; 'LC-31 indicates a 3.3V part
  - The -27, -33, -40, etc. extensions indicates the maximum clock frequency in MHz

NO.			'C31-27		'C31-33 'LC31-33		'C31-40 'LC31-40		'C31-50		'C31-60		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	
12	$t_{d(H1L-SL)}$	Delay time, H1 low to $\overline{STRB}$ low	0‡	13	0‡	10	0‡	6	0‡	5	0‡	5	ns
13	$t_{d(H1L-SH)}$	Delay time, H1 low to $\overline{STRB}$ high	0‡	13	0‡	10	0‡	6	0‡	5	0‡	5	ns
14	$t_{d(H1H-RWL)R}$	Delay time, H1 high to R/$\overline{W}$ low (read)	0‡	13	0‡	10	0‡	9	0‡	7	0‡	6	ns
15	$t_{d(H1L-A)}$	Delay time, H1 low to A valid	0‡	16	0‡	14	0‡	11	0‡	9	0‡	8	ns
16	$t_{su(D-H1L)R}$	Setup time, D before H1 low (read)	18		16		14		10		9		ns
17	$t_{h(H1L-D)R}$	Hold time, D after H1 low (read)	0		0		0		0		0		ns
18	$t_{su(RDY-H1H)}$	Setup time, $\overline{RDY}$ before H1 high	10		8		8		6		5		ns
19	$t_{h(H1H-RDY)}$	Hold time, $\overline{RDY}$ after H1 high	0		0		0		0		0		ns
20	$t_{d(H1H-RWH)W}$	Delay time, H1 high to R/$\overline{W}$ high (write)		13		10		9		7		6	ns
21	$t_{v(H1L-D)W}$	Valid time, D after H1 low (write)		25		20		17		14		12	ns
22	$t_{h(H1H-D)W}$	Hold time, D after H1 high (write)	0		0		0		0		0		ns
23	$t_{d(H1H-A)W}$	Delay time, H1 high to A valid on back-to-back write cycles (write)		23		18		15		12		10	ns
24	$t_{d(A-RDY)}$	Delay time, $\overline{RDY}$ from A valid		10‡		8‡		7‡		6‡		6‡	ns

‡ This value is characterized but not tested

Time Domain Definition of TMS32C031-50 Read Operation

- 12 Delay time, H1 low to STRB* low (5 ns MAX)
- 15 Delay time, H1 low to A valid (9 ns MAX)
- 24 Delay time, RDY* from A valid (6 ns characterized, not tested)
- 18 Setup time, RDY* before H1 high (6 ns MIN)
- 19 Hold time, RDY* after H1 high (0 ns MIN)
- 16 Setup time, D before H1 low (read, 10 ns MIN)
- 17 Hold time, D after H1 low (read, 0 ns MIN)



NOTE A: STRB remains low during back-to-back read operations.

Incorporating Time Domain Constraints into the Design Example

- Our next step is determining if the interface design satisfies the performance requirements of the processor
- For the time being, we'll only be concerned with reading from the 8255A Programmable Peripheral Interface
- There are two timing parameters that must be satisfied by the interface and the 8255A
  - The RDY* signal must be generated by the interface prior to the rising edge of H1 to avoid adding a wait state
    - This includes all combinational delays through the interface as well as satisfying the setup and hold times for sampling around H1
  - The DATA must be on the bus and valid before the next falling edge of H1 at which time the data is sampled by the processor
    - Once again, this will include propagation delays through the interface and through the appropriate path within the peripheral and the satisfaction of the setup and hold requirements around the falling edge of H1

Incorporating Time Domain Constraints into the Design Example

- We'll look at generation of the RDY* signal first and assume we're using the 50MHz version of the DSP
  - This yields a system clock period of 20 ns and an H1/H3 clock period of 40 ns
- The first timing parameters to look at are the generation of the valid address bits since the RDY* is generated from address bits A23 – A20.
  - The table indicates that the worst case delay (MAX) between H1 going low and ADDRESS valid (timing parameter 15) is 9 ns
  - The next relevant delay is the combinational propagation delay through the address decoder and the enabling of the tristate output to generate the RDY* signal
    - We'll assume this is the value to be determined, i.e., the interface must meet this requirement for it to operate correctly with the processor

Incorporating Time Domain Constraints into the Design Example

- Finally, we have to satisfy the setup time of of the RDY* signal with respect to the rising edge of H1 (timing parameter 18)
  - The table defines this as worst case (MIN) of 6 ns
- Since the delay from H1 going low to H1 going high is 20 ns, the maximum propagation delay through the interface is:
  - 20 ns – 9ns (H1 low to valid A) – 6 ns (setup time) = 5 ns
  - If this requirement is not met, either the interface needs to be redesigned (using different hardware) or a wait state needs to be inserted
- The last piece of the analysis would be satisfying the hold time (19) of the RDY* signal after H1 going high but since this is specified as 0 ns, it is correct by construction
  - In reality, hold times are often negative but are always specified as 0 ns in that case

Incorporating Time Domain Constraints into the Design Example

- The next critical timing parameter is valid data being available on the data bus on the next rising edge of H1
  - From the DSP side of the interface, data is sampled on the falling edge of H1 so the timing requirement to be met is:
    - H1 period – setup time (DATA valid to H1 going low)
    - At 50 MHz, the period of H1 is 40 ns and the Data valid setup time before H1 going low is 10 ns (timing parameter 16)
    - This allows 30 ns for the combinational delay of generating (worst case) CS* or RD* plus the internal delay in the 8255A
  - On the 8255A side of the interface, recall that
    - the CS* signal is generated from address bits A23 – A20 and
    - the RD* and WR* signals are generated from DSP signals STRB* and R/W*

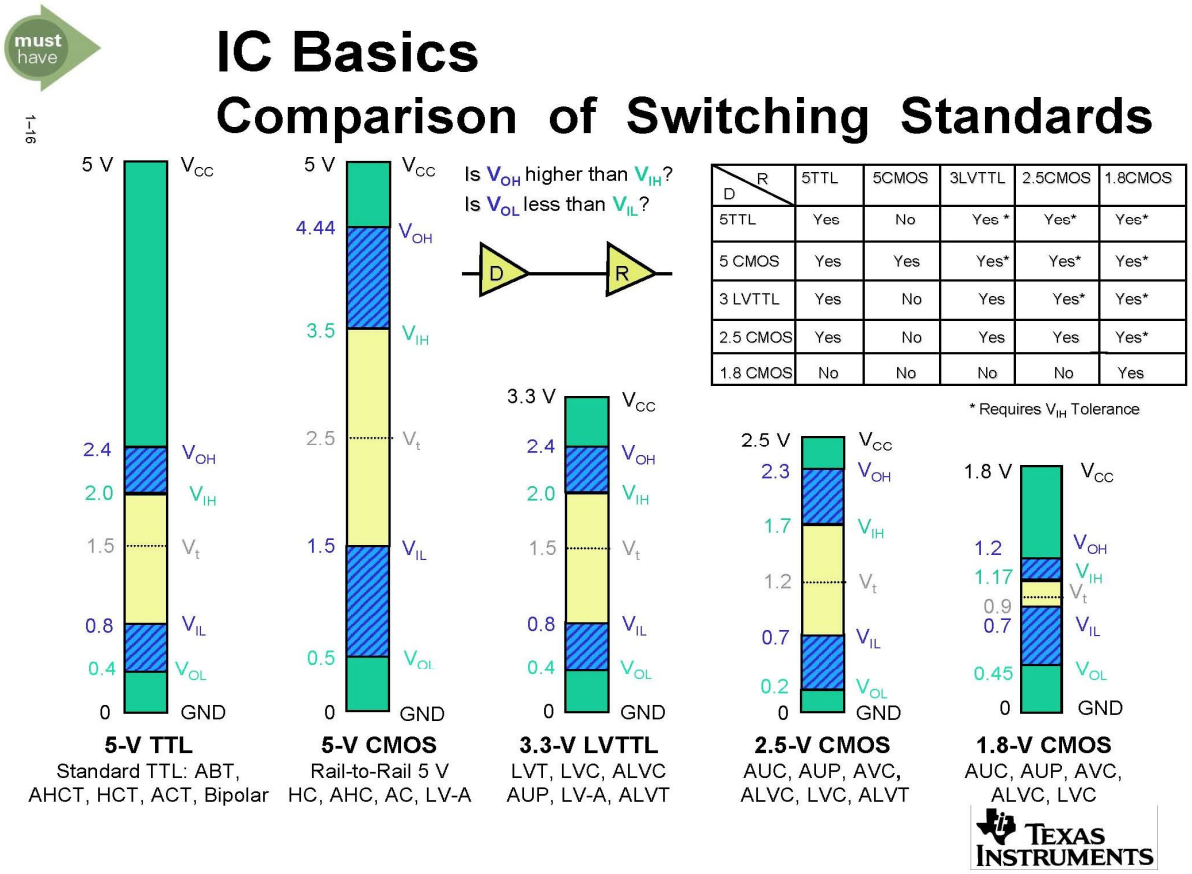
Incorporating Time Domain Constraints into the Design Example

- Worst case propagation delay to generate control signals, is longest of:
  - (12) 5 ns MAX H1 low to STRB* (for RD* and WR*)
  - (15) 9 ns MAX H1 low to valid Address (for CS*)
  - If we assume the propagation delays through the interface are roughly equivalent, the generation of CS* will be the critical path to the inputs of the 8255A
- The setup time for Data becoming valid before H1 going low (timing parameter 16) is 10 ns and thus the worst case delay through the 8255A is:

40 ns (H1 period) – 9 ns (H1 to A valid) - 5 ns (max delay to generate CS*)
– 10 ns (setup time, D before H1 low) =

16 ns worst case delay through 8255A critical path (propagation delay)

Electrical Requirements



R / D	5TTL	5CMOS	3LVTTL	2.5CMOS	1.8CMOS
5TTL	Yes	No	Yes *	Yes*	Yes*
5 CMOS	Yes	Yes	Yes*	Yes*	Yes*
3 LVTTL	Yes	No	Yes	Yes*	Yes*
2.5 CMOS	Yes	No	Yes	Yes	Yes*
1.8 CMOS	No	No	No	No	Yes

* Requires V_{IH} Tolerance

Level Shifting

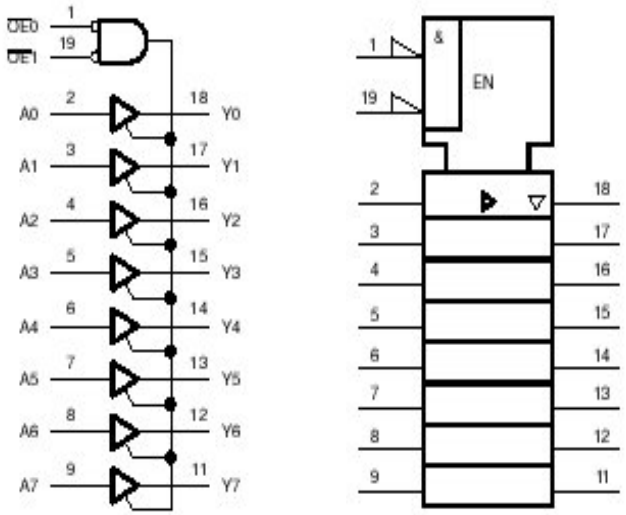
- Sometime level shifting is necessary to allow different logic families to communicate, sometimes it is not necessary
- Some (many) contemporary processors operate at 3.3 Volts but their inputs can accept (tolerate) outputs from 5 Volt devices
- Use of inverters to change levels when necessary
  - Use a pair of TTL inverters to boost 3.3 Volt levels to 5 Volt levels

Interface to External Board

- E153BSYS board interfaces to user board (E153PLGIN) through EUROcard connector
- Physical vs. Logical Interfacing
  - Sometimes interface includes both
- Electrical Isolation
  - Public Networks
    - Telephone, etc.
  - Card Cages
  - Backplanes
  - Daughter boards

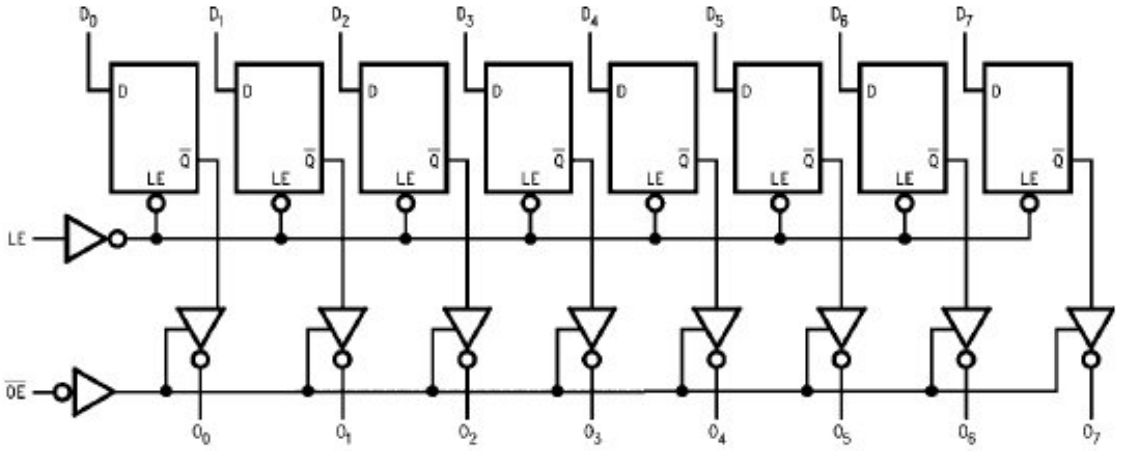
74541 Octal Tristate Buffer

- Buffers with tri-state controls
  - available in many logic families)
    - LS, ALS, C, HC, ACT, HCT, ABT, …
  - Version with inverting buffers is 74540



74573 Octal Latch with Tri-state

- Latch with tri-state controls (available in many logic families)
- Octal latch that enables Di to Qi while CLK high (latches when CLK low). There is a common output enable control line.



74245 Octal Transceiver

- Bi-directional bus switch (available in many logic families)
  - 74ACT245 used on E153BSYS board address and data buses