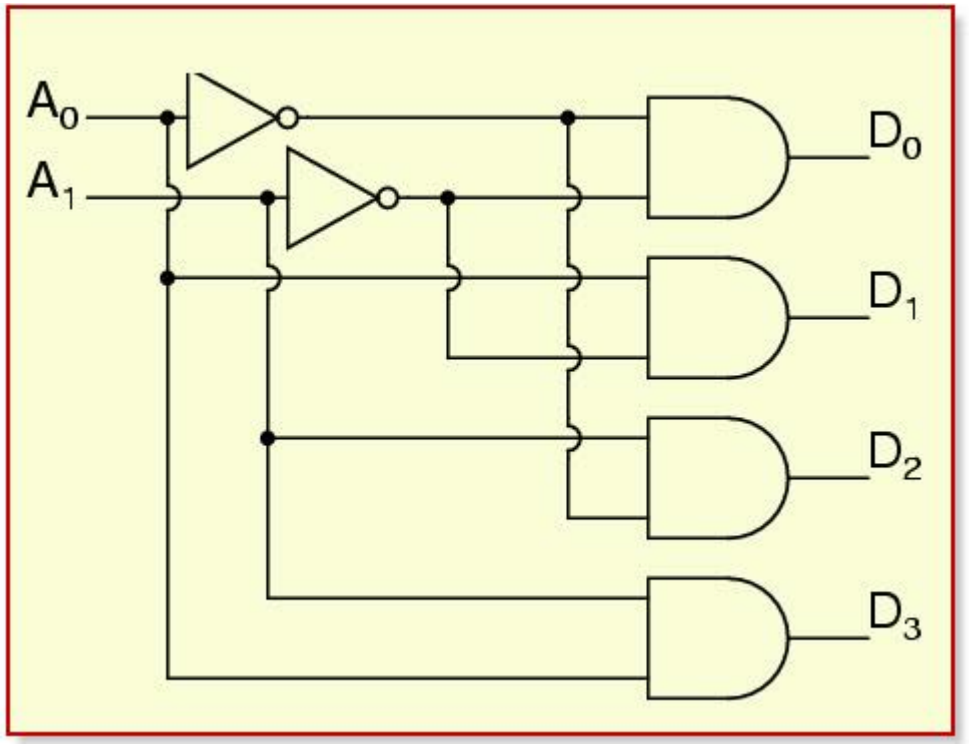
Chapter 20 – Intro to Interrupts

CPSC 240-09
John Overton

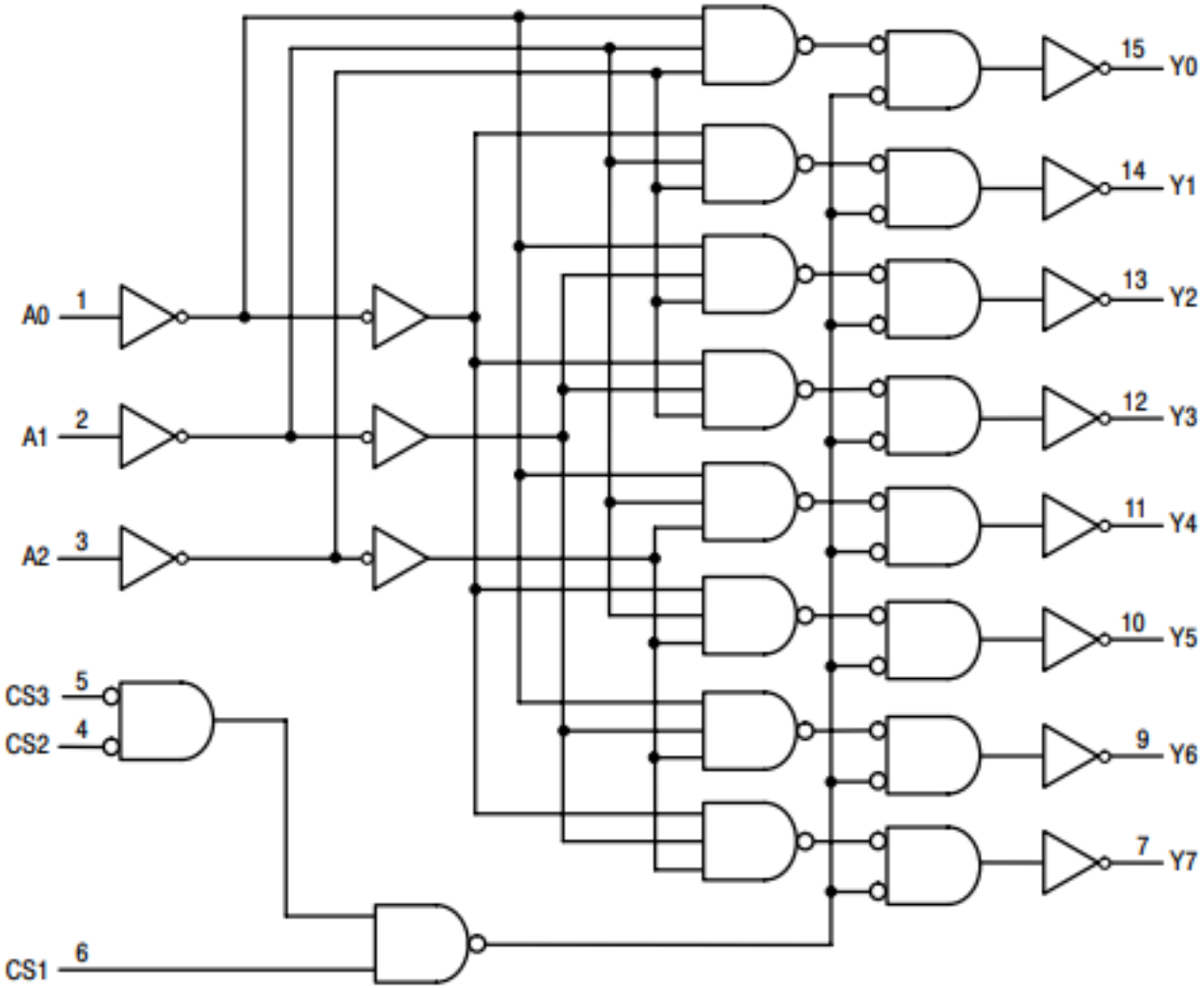
Review

Decoders

Input		Output			
A_1	A_1	D_0	D_1	D_2	D_3
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0



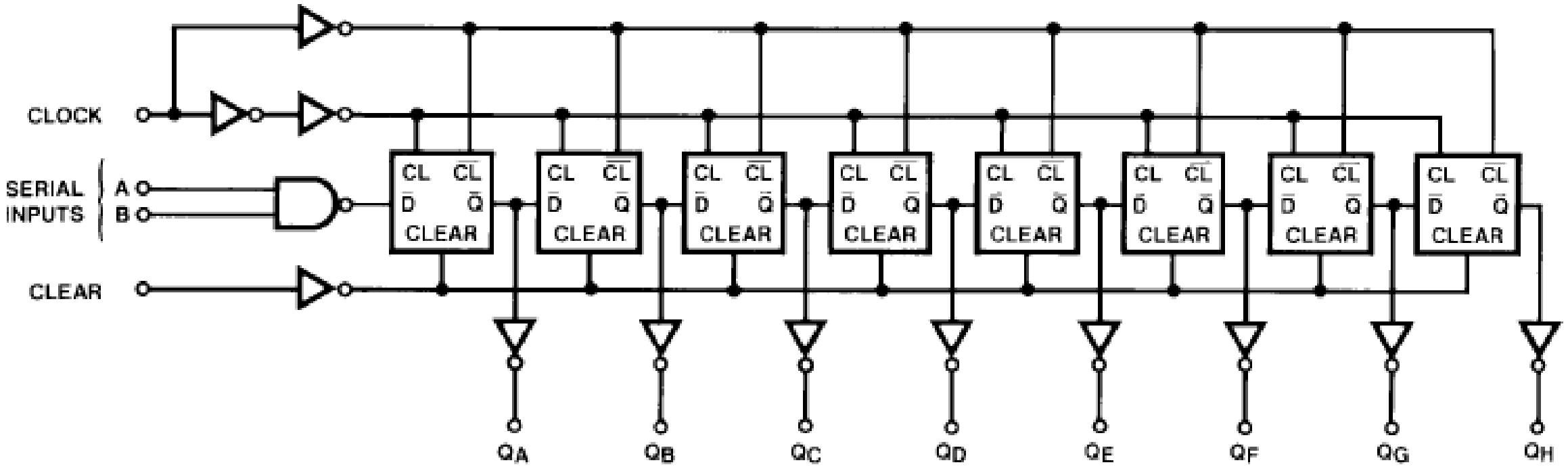
Decoders – 74HC138



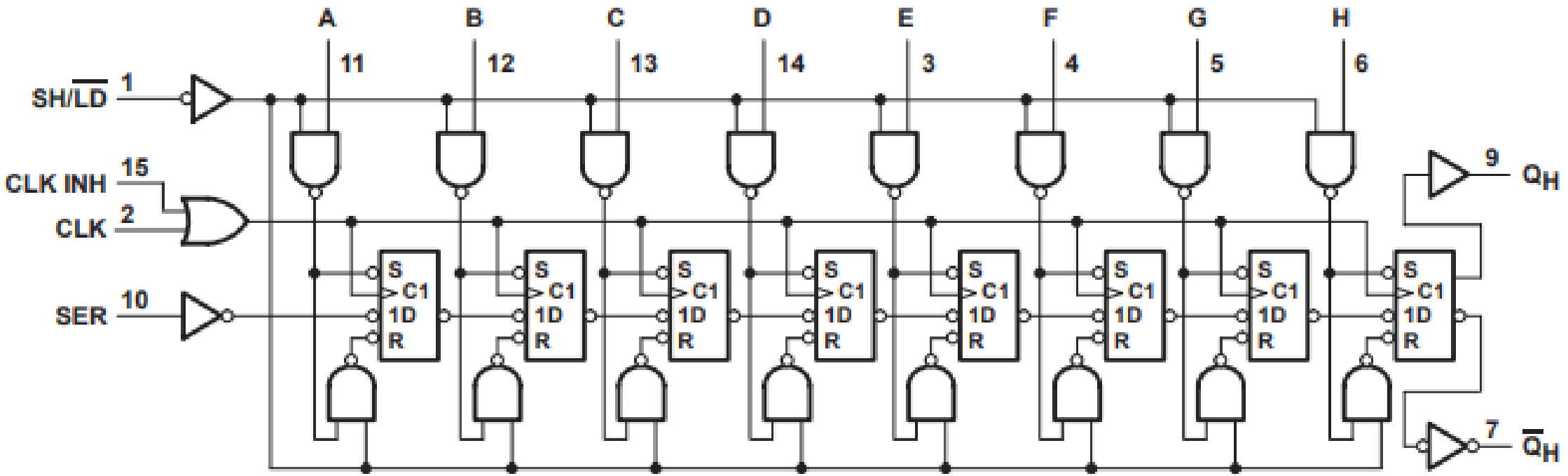
Inputs						Outputs							
CS1	CS2	CS3	A2	A1	A0	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	X	H	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	L	H	H	H	H	H	L	H	H	H	H
H	L	L	H	L	L	H	H	H	H	L	H	H	H
H	L	L	H	L	H	H	H	H	H	H	L	H	H
H	L	L	H	H	L	H	H	H	H	H	H	L	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L

H = High Level (1) L = Low level (0)
X = Don't Care
"Active Low"
(The 74HC238 has Active High I/O)

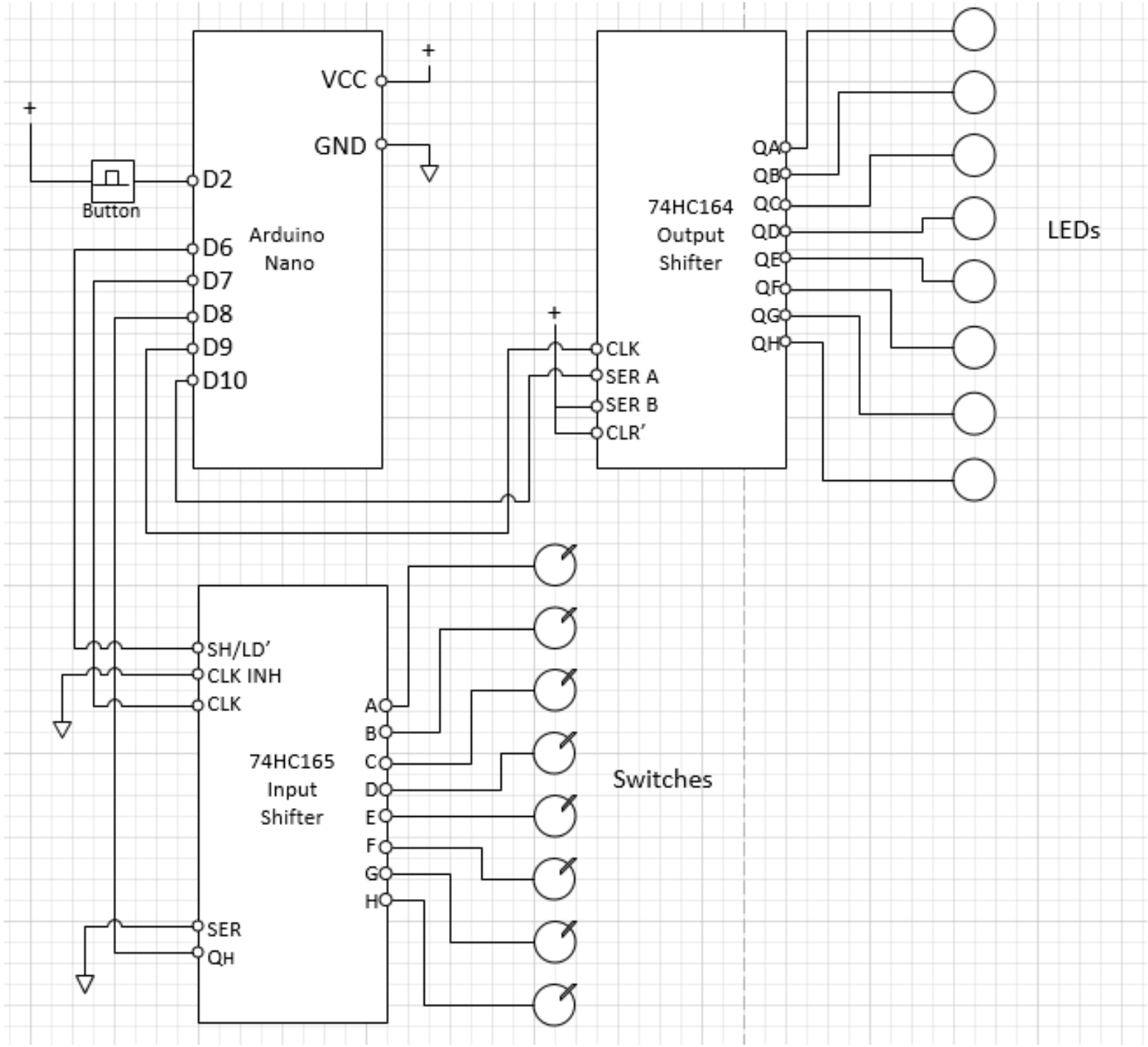
Review – 74HC164 – Serial In & 8 Out Shift Register



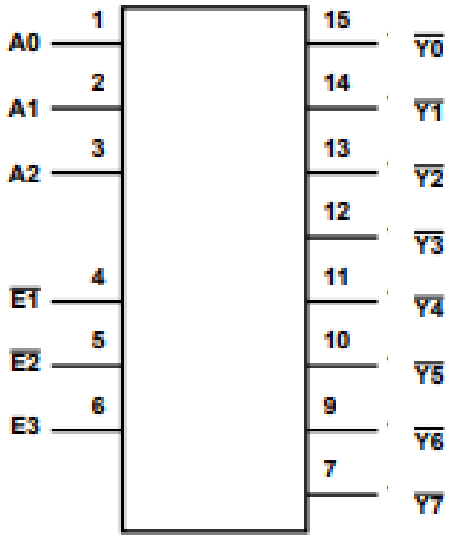
Review – 74HC165 – 8 In and Serial Out Shift Register



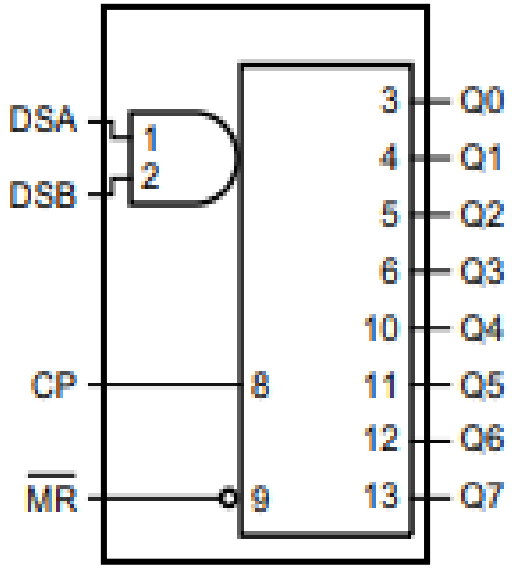
Example using shift registers



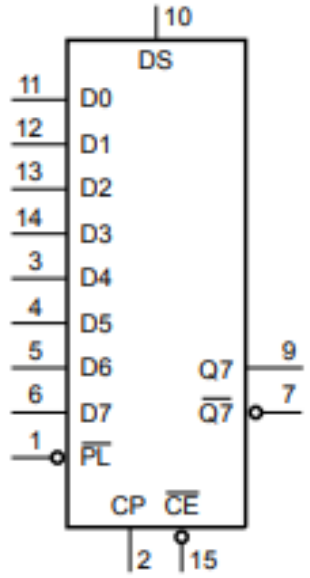
Common Blocks When using in Schematics



74HC138

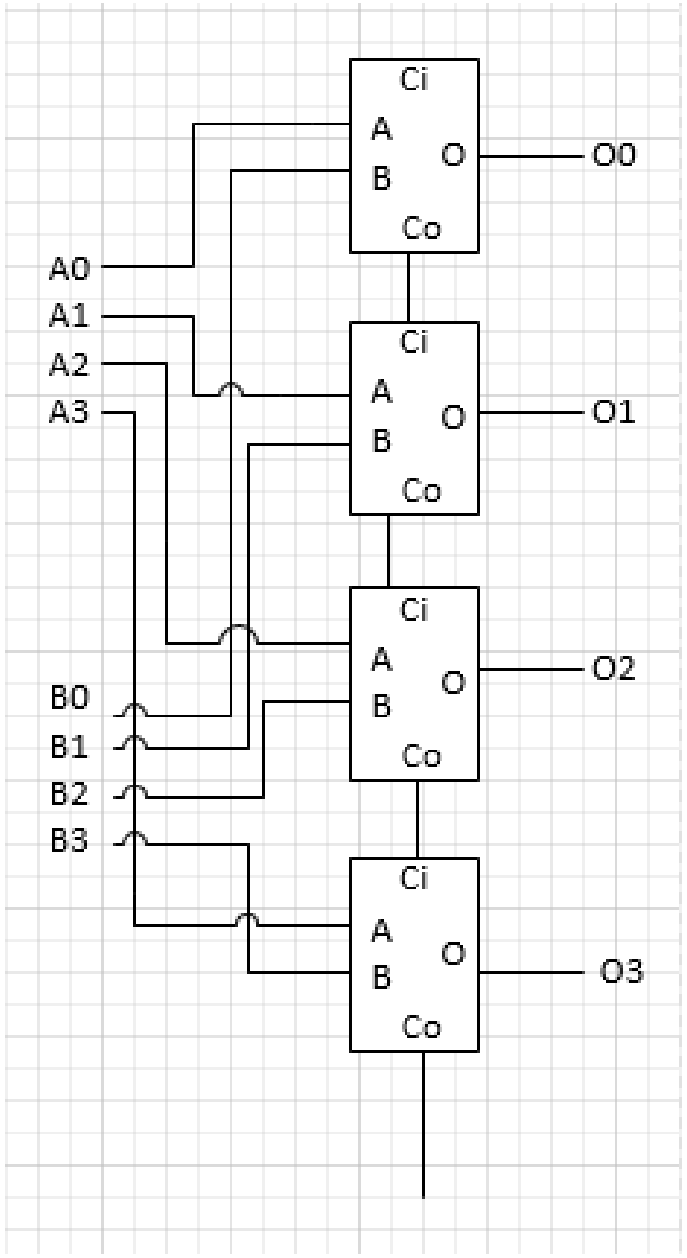


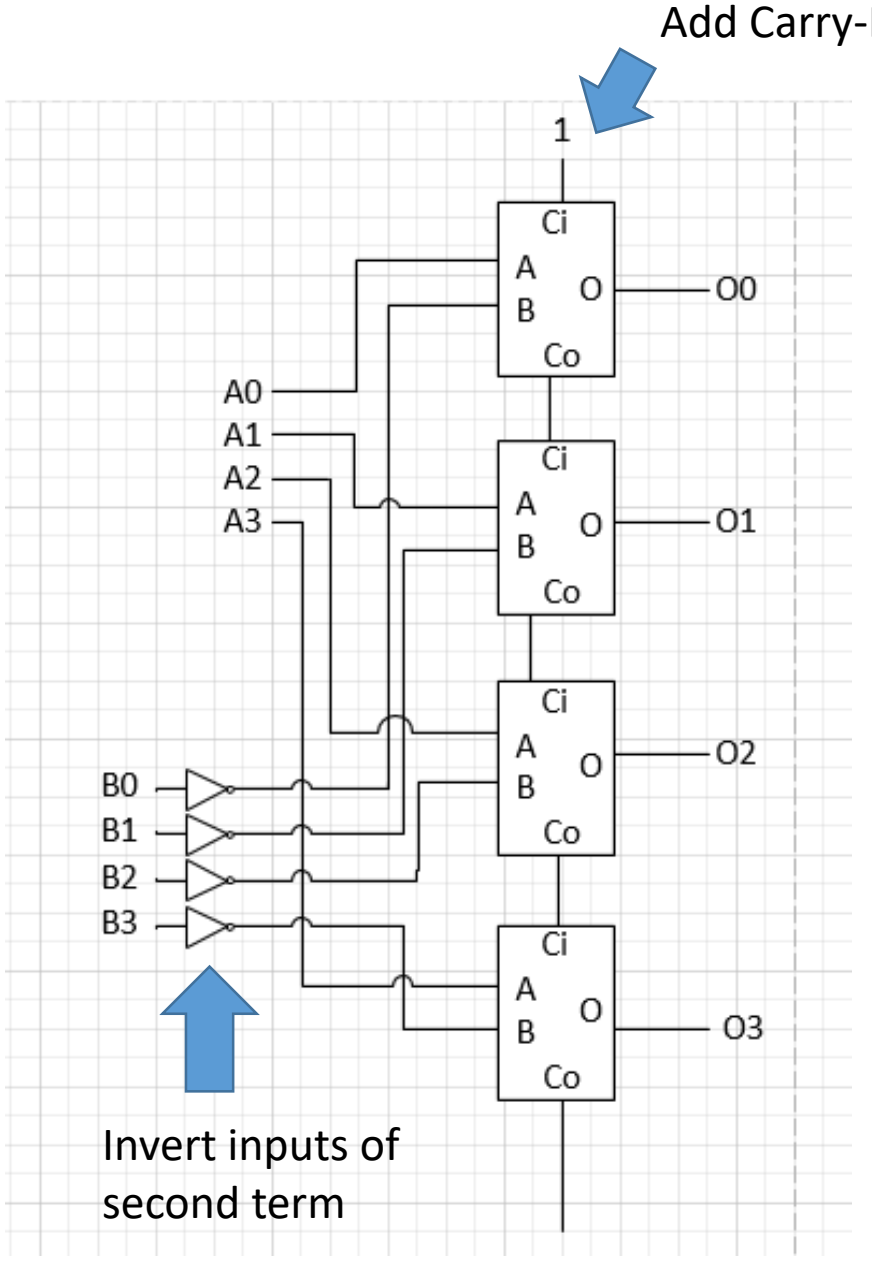
74HC164



74HC165

Turning an adder into a subtractor: 2's Compliment





Interrupts

Interrupts

- Interrupts cause the computer to pause what it is currently doing and handle some very important, but short interruption before going back to what it was doing before
- Interrupts can be hardware related or can be software related
- Modern computers can handle 100’s, 1000’s, even millions of interrupts in a second, if needed
- Some interrupts are bad, such as program exceptions (these fall into the software interrupts category)
- Many processor architectures have a software interrupt, which programs use to ask the computer to provide a service for it (in x86_64, the INT instruction has been replaced by the syscall instruction)

Software Interrupts

- Programs cause interrupts all the time. A page fault is a type of software interrupt (of sorts)
- When a software interrupt is not expected, i.e., caused by a software problem, then we call it a Program Exception
- A SIGFAULT (which is also a type of page fault, but usually happens when you have a bad pointer) is a software interrupt and is an exception
- A DIVIDE BY ZERO is a software interrupt and is an exception

Hardware Interrupts

- I/O devices (keyboards, network adapters, disk drives, USB devices, etc)
- Interval timers - There's usually more than one. A system timer provides a constant "tick" interrupt periodically. Another timer is used to notify a program after a requested interval time has concluded
- Other CPUs (on multiprocessor systems)

More about Interrupts

- On many processor architectures, such as the x86_64, there are various privilege levels that instructions can run at. For example, an OS will run at level 0, a “user” program will run at level 3.
- Similarly, interrupt code can run at any of these four levels
- However, in Linux, only 2 levels are used, 0 and 3
- When an interrupt occurs, code that runs due to the interrupt is called an Interrupt Service Routine (ISR)
- Generally, the operating system must set up the address of an ISR. In the x86_64 architecture, the Interrupt Descriptor Table (IDT) is a table of ISRs
- When an interrupt occurs, the processor saves the current RIP register, then loads the appropriate ISR address into the RIP register to being executing the ISR.

More about Interrupts

- When the ISR starts running, it must save the current “context” of the processor so that the processor can restart where it left off after the ISR is finished
- In most processor architectures (software implementation of the Operating System), the ISR is divided into two parts: the first-level interrupt handler (FLIH) and the second-level interrupt handler (SLIH). These are also called the top-half and the bottom-half
- The FLIH usually saves the context, then takes care of any hardware requirements (such as acknowledging the interrupt, resetting hardware and saving any information that may only be available at the time of the interrupt. The FLIH is usually common to many different types of interrupts
- The SLIH is code that is more specific to an interrupt, such as a disk drive interrupt which may need to schedule the next I/O request to the disk drive

Polling Vs interrupts

- In polling, the CPU keeps on checking all the hardware of the availability of any request
- Polling is like standing at the window waiting for someone to come to your door
- The CPU spends wasted time waiting/checking for something to happen

- With Interrupts, the CPU takes care of the hardware only when the hardware requests for some service
- An interrupt is like a doorbell
- The CPU does not waste time waiting/checking, so it can do other things (like running programs)

Assignment

- Use one of the online simulators to see how various flip-flops work:
  1. Look at Chapter 19 assignment on page 286 of the book - Suggested Projects #1