Chapter4: Combinational Logic

Lecture6- Design Comparator Circuit, Introduction to Decoders

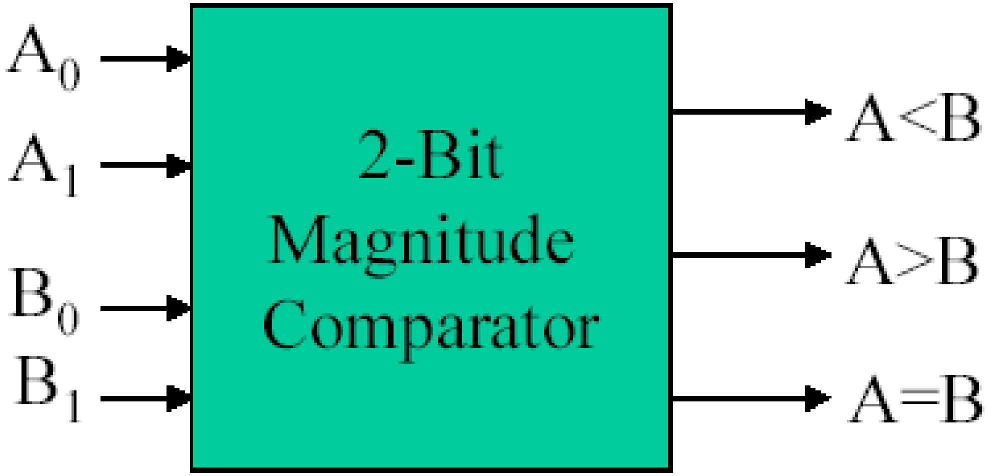
Engr. Arshad Nazir, Asst Prof
Dept of Electrical Engineering
SEECS

Objectives

- Design Magnitude Comparator
- Introduction to Decoders

Magnitude Comparator

- The comparison of two numbers is an operation that determines if one number is greater than, less than or equal to the other number
- A magnitude comparator is a combinational circuit that compares two numbers, A and B and determines their relative magnitudes
- The outcome of the comparison is specified by three binary variables that indicates whether A>B, A=B or A<B



Magnitude Comparator

- If we follow the traditional design approach of truth table then comparing two n-bit numbers will have 2^{2n} entries in the truth table and becomes too complicated for large values of n.
- However, a comparator circuit possess a certain amount of regularity. Digital functions that possess an inherent well defined regularity can usually be designed by means of algorithmic procedure. This reduce design efforts and reduce human errors
- An algorithm is a procedure that specifies a finite set of steps, if followed, give the solution to a problem. The algorithm is direct application of the procedure a person uses to compare the relative magnitude of two numbers

Developing Algorithm

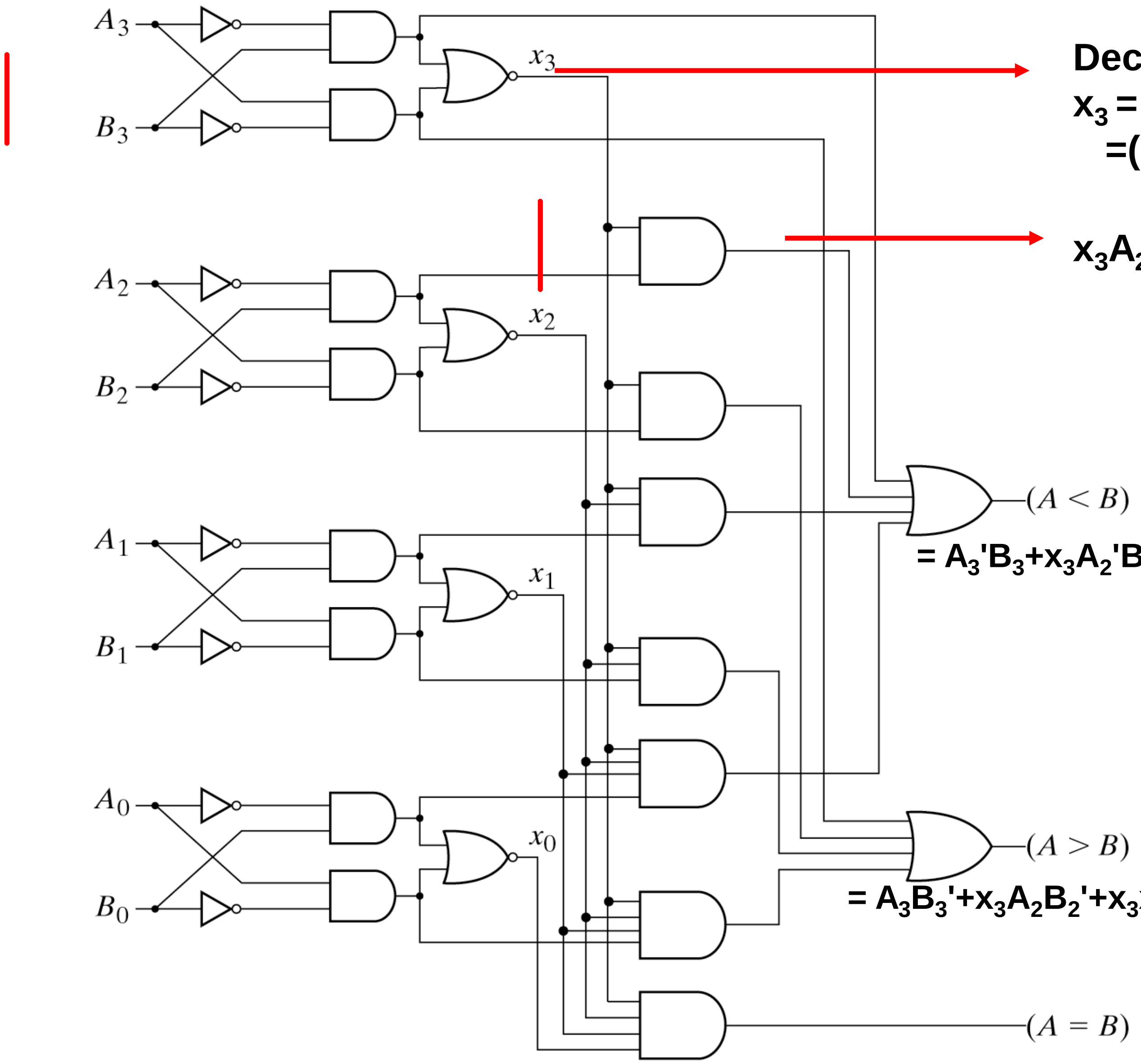
- Consider the two numbers A and B, with four digits each, the coefficients of numbers with descending significance can be written as:
  - $A = A_3A_2A_1A_0$ and $B = B_3B_2B_1B_0$
- The two numbers are equal if all pairs of significant digits are equal:
  - $A=B$ if $A_3=B_3$, $A_2=B_2$, $A_1=B_1$and $A_0=B_0$
- For binary (either 1 or 0) digits the equality relation of each pair of bits can be expressed logically with exclusive-NOR function to test if $A_i = B_i$ as
  - $x_i = (A_i \oplus B_i)' = (A_iB_i'+A_i'B_i)' = A_iB_i+A_i'B_i'$ for $i = 0,1,2,3$
  - $x_i = 1$ only if the pair of bits in position i are equal otherwise $x_i = 0$
- Therefore we can check if A = B by
  - $(A=B) = x_3x_2x_1x_0$
- The symbol (A=B) is binary output variable that is equal to 1 only if all pair of digits of the two numbers are equal

Developing Algorithm

- To determine if A>B or A<B, we inspect the relative magnitudes of pairs of significant digits starting from the most significant position. If the two digits are equal, we compare the next lower significant pair of digits. This comparison continues until a pair of unequal digits is reached
- If the corresponding digit of A is 1 and that of B is 0, we conclude that A>B. If the corresponding digit of A is 0 and that of B is 1, we conclude that A<B
- The sequential comparison can be expressed logically by the two Boolean functions
  - $(A>B) = A_3B_3'+x_3A_2B_2'+x_3x_2A_1B_1'+x_3x_2x_1 A_0B_0'$
  - $(A<B) = A_3'B_3+x_3A_2'B_2+x_3x_2A_1'B_1+x_3x_2x_1 A_0'B_0$
- The symbols (A>B) and (A<B) are binary output variables that are equal to 1when A>B or A<B respectively

Developing Algorithm

- The gate implementation for a magnitude comparator involves a certain amount of repetition so it is simpler than it seems.
- The unequal outputs can use the same gates that are needed to generate the equal output
- The logic diagram of the 4-bit magnitude comparator is shown in Figure 4 – 17
- The four x outputs are generated with exclusive-NOR circuits and applied to an AND gate to give the output binary variable (A=B)
- The other two outputs use the x variable to generate the Boolean functions
- The procedure for obtaining magnitude comparator circuits for binary numbers with more than four bits is obvious from the above steps



Decides if A3=B3

$x_3 = (A_3B_3'+A_3'B_3)'$

$=(A_3 \oplus B_3)'$

$x_3A_2'B_2$

$= A_3'B_3+x_3A_2'B_2+x_3x_2A_1'B_1+x_3x_2x_1 A_0'B_0$

$= A_3B_3'+x_3A_2B_2'+x_3x_2A_1B_1'+x_3x_2x_1 A_0B_0'$

Fig. 4-17 4-Bit Magnitude Comparator

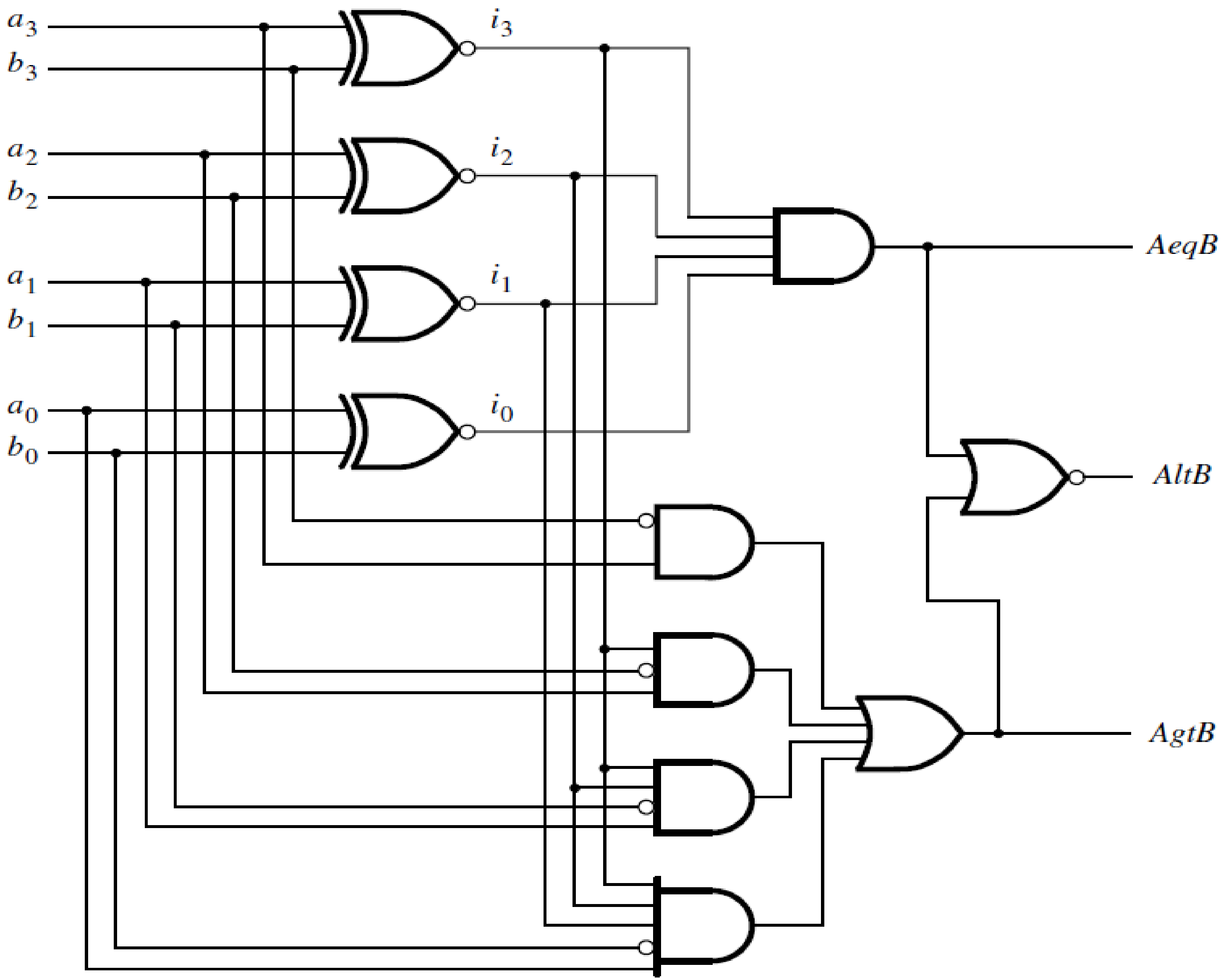
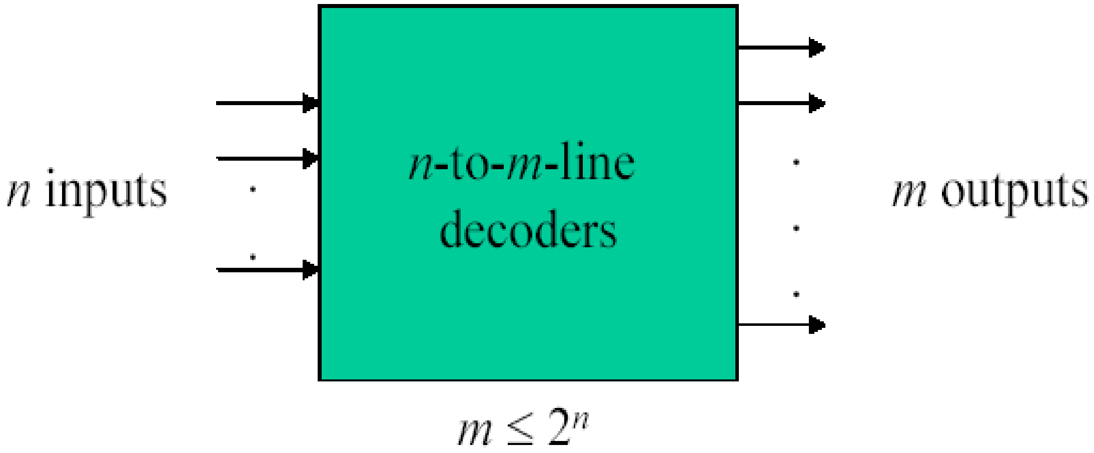


Figure 4.22 A four-bit comparator circuit.

Decoders

- A decoder is a combinational circuit that converts binary information from n input lines to a maximum of 2^n unique output lines. Only one output can be active (high) at any time.
- Decoders are a class of combinational logic circuits that convert a set of input variables representing a code into a set of output variables representing a different code. The relationship between the input and output codes can be expressed in a truth table i.e 4-to-10-lines decoder circuit.
- If the n-bit coded information has unused combinations, the decoder has fewer than 2^n output-



Decoder Applications

- Generate minterms/complement of minterms and are used for functions implementation.
- Memory addressing- Decoders are widely used in the memory system of a computer where they respond to the address code generated by the central processor to activate a particular memory location.
- Code Conversion. Example is BCD-to-7-segment decoder.
- DeMUX function.
- Used in conjunction with counters to decode (detect) counter states and provide timing or sequencing signals.
- Provide enabling inputs when used in the design of MUXs with tri-state gates.
- Computers communicate with peripheral devices (printers, modems, scanners, keyboards, video monitors, external disk drives and other computers) by sending and/or receiving data through I/O ports. Decoders are used to select I/O as determined by the computer to receive or send data.

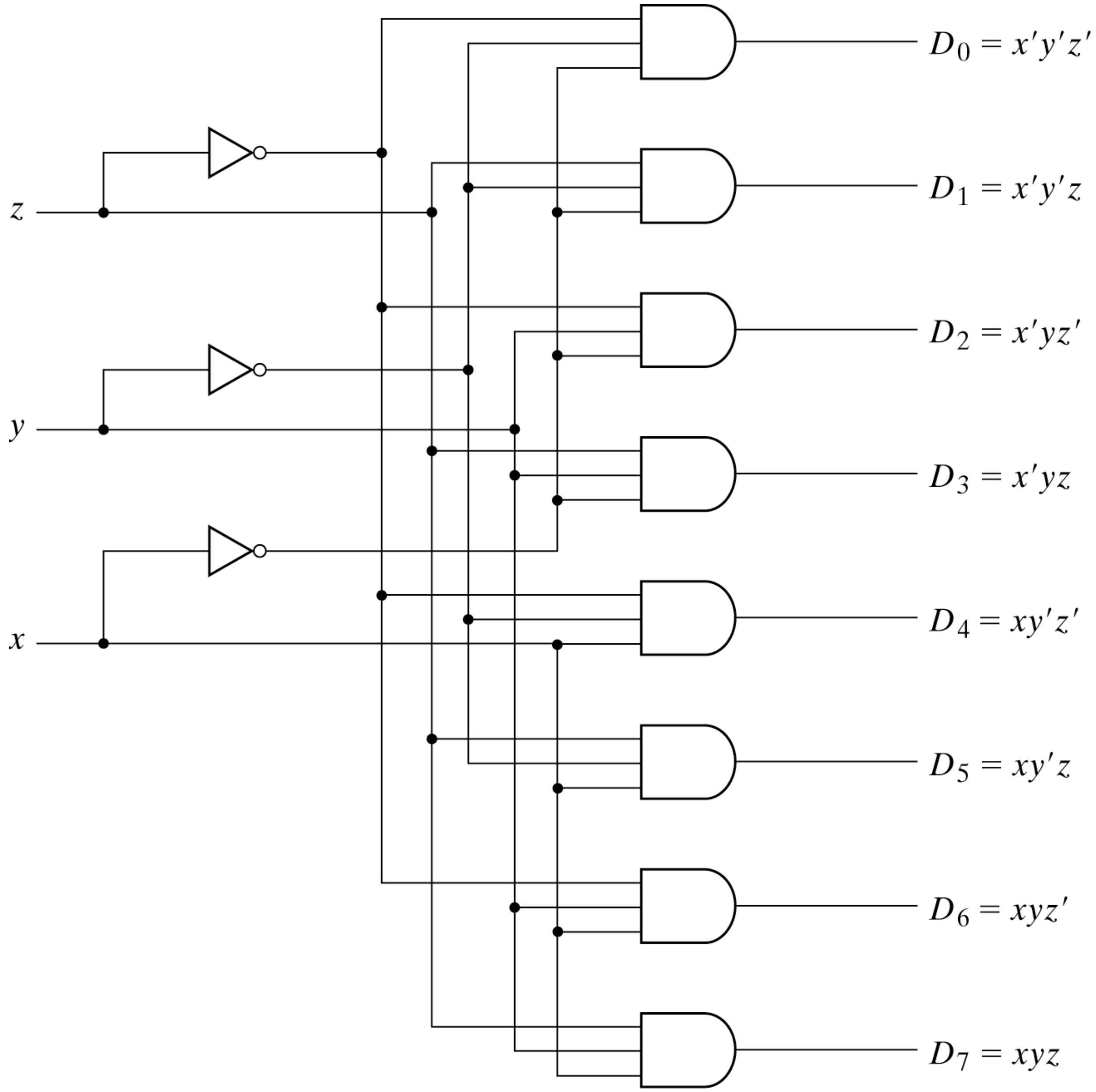


Fig. 4-18 3-to-8-Line Decoder

Implementation of Full Adder with a Decoder

- There are three inputs and eight outputs so we need 3-to-8-line decoder
- Two OR gates are required for logical sum of the desired minterms

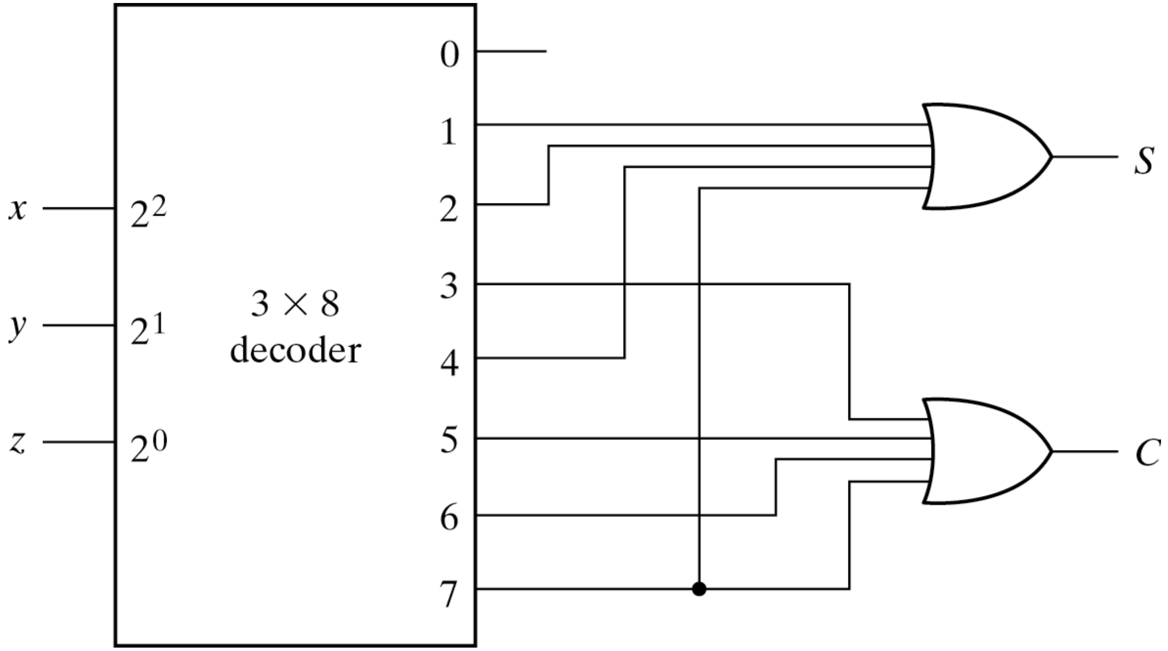
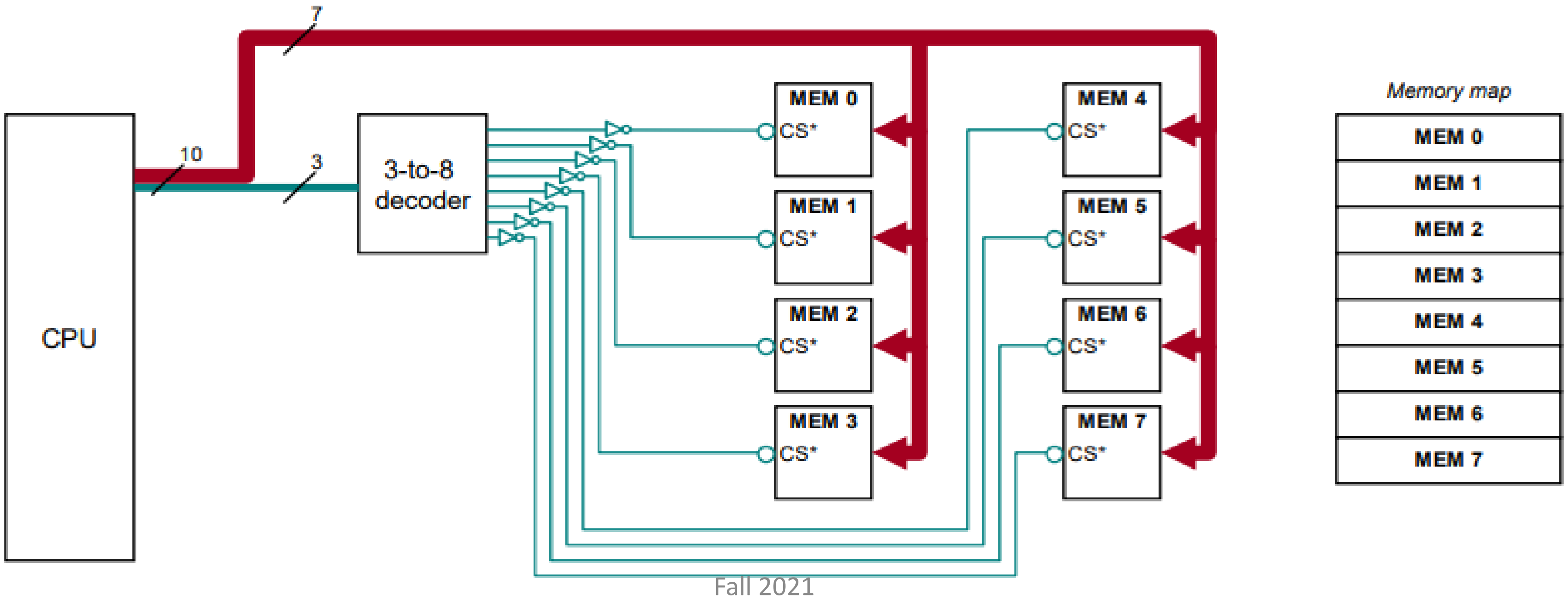


Fig. 4-21 Implementation of a Full Adder with a Decoder

A very simple example

- **Let’s assume a very simple microprocessor with 10 address lines (1KB memory)**
- **Let’s assume we wish to implement all its memory space and we use 128x8 memory chips**
- **SOLUTION**
  - We will need 8 memory chips (8x128=1024)
  - We will need 3 address lines to select each one of the 8 chips
  - Each chip will need 7 address lines to address its internal memory cells

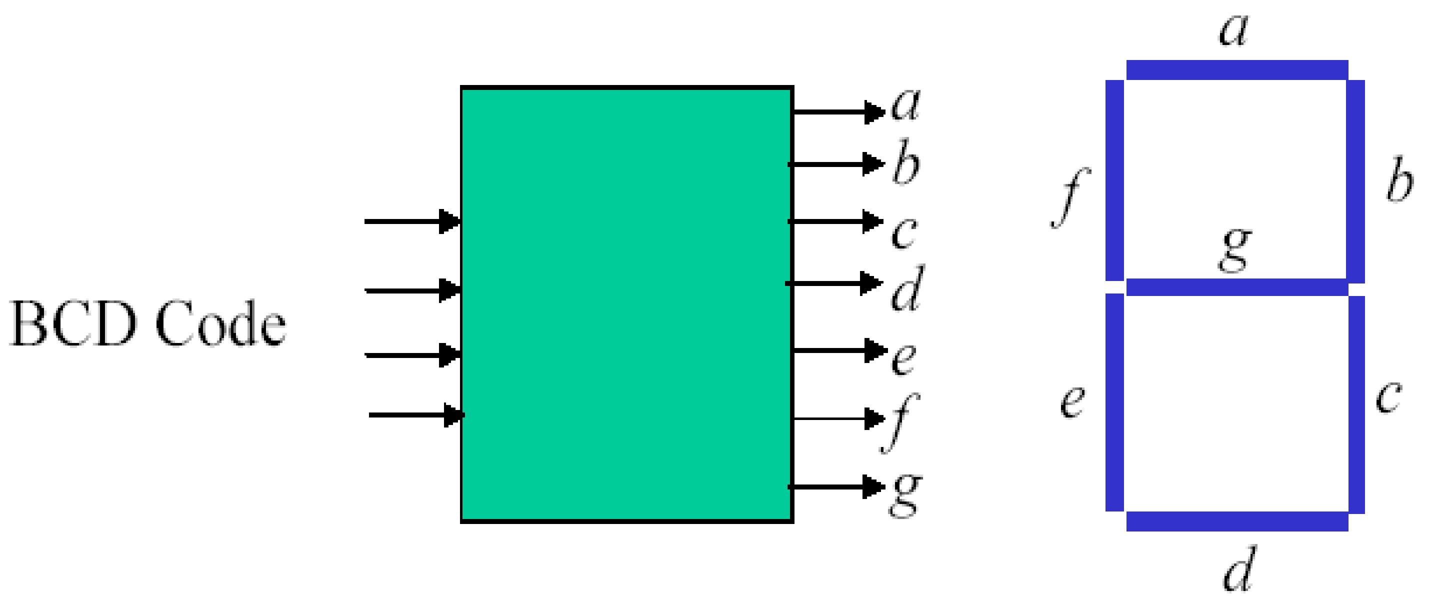


Memory map

MEM 0
MEM 1
MEM 2
MEM 3
MEM 4
MEM 5
MEM 6
MEM 7

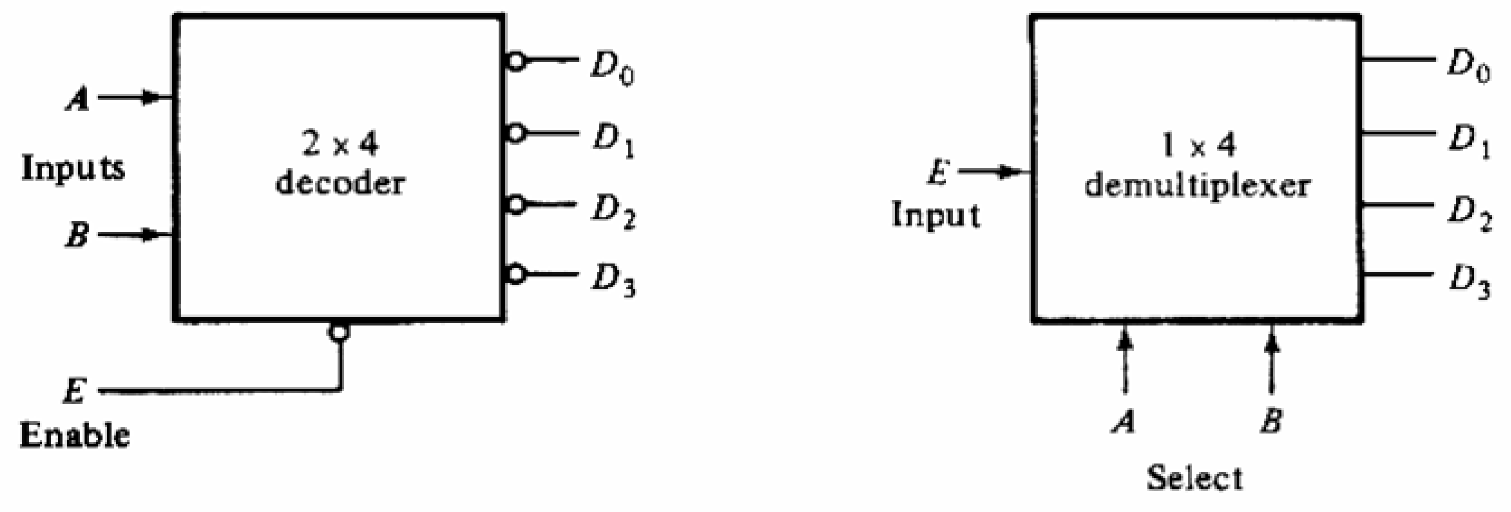
Decoder Example (Code Converter)

- BCD-to-seven-segment display converter is one common example of code converters, which converts one BCD digit into information suitable for driving a digit-oriented display.



Demultiplexer

- A decoder with an enable input (Figure 4-19) can function as demultiplexer (1-to-4-line demultiplexer)
  - E is taken as data input line and A and B are takes as selection inputs

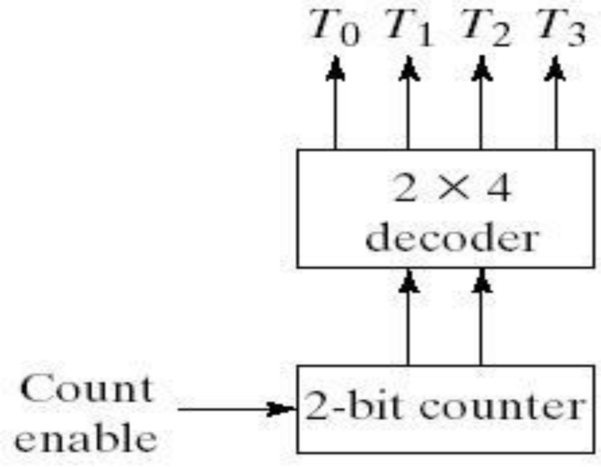


(a) Decoder with enable

(b) Demultiplexer

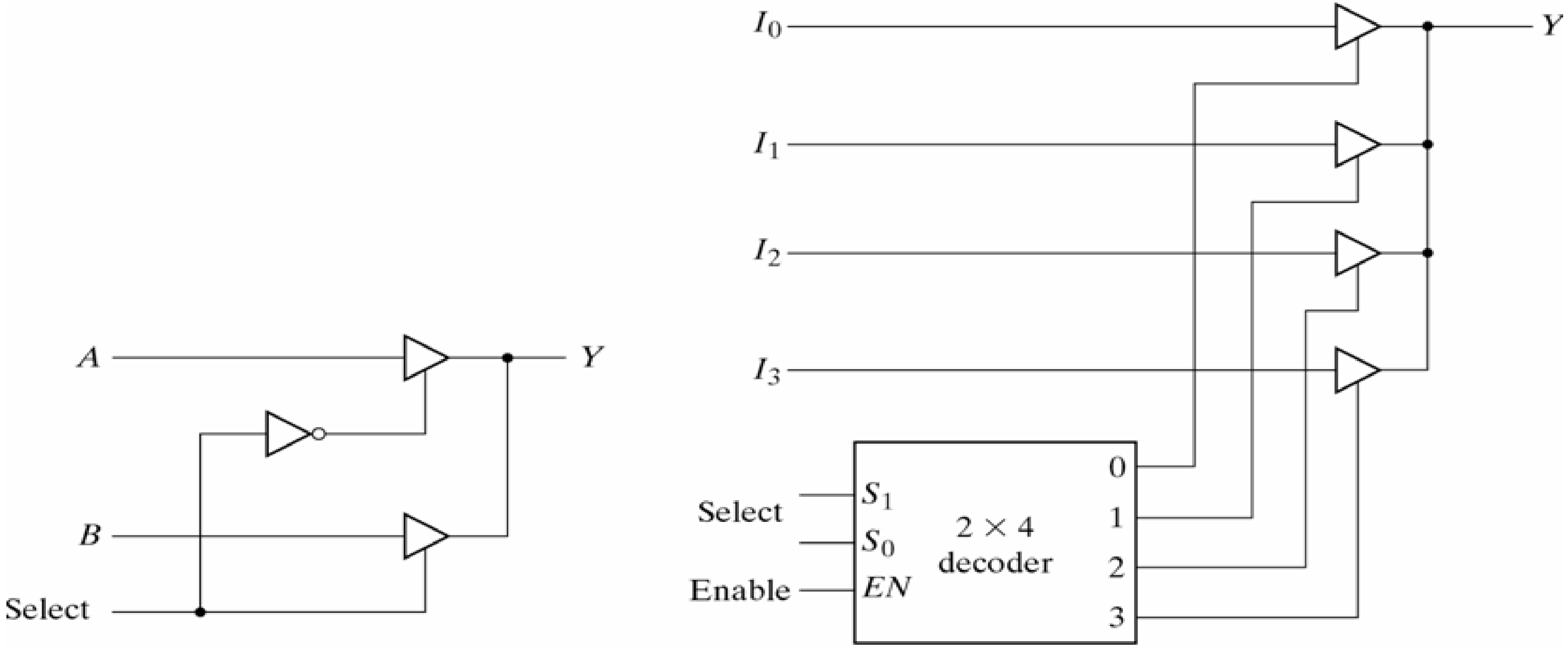
Timing Signals Generation using Counters

- Counters may be used to generate timing signals to control the sequence of operations in a digital system. 2^n timing signals can be generated using an *n*-bit binary counter together with an *n*-to-2^n-line decoder



(b) Counter and decoder

MUX Design using Tri-State gates and Decoders



(a) 2-to-1- line mux

(b) 4 - to - 1 line mux

Fig. 4-30 Multiplexers with Three-State Gates

The End