Comp 182 Chapter 3
Digital Logic Level

Gates

And, Or, Not

Nand, Nor

How they are created.

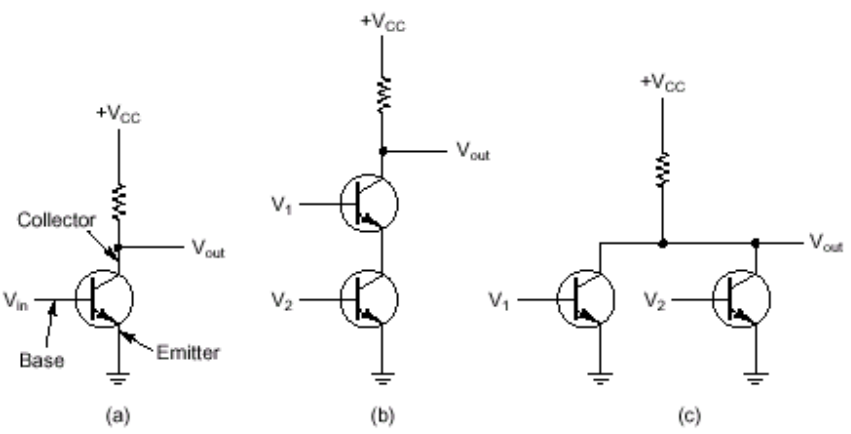


Figure 3-1. (a) A transistor inverter. (b) A NAND gate. (c) A NOR gate.

The inverter is a Not. Typically the Not, Nand, Nor gates are used as the basic building blocks because they can be created with fewer transistors.

If an and gate is required - for example - it can be created by tying a a Nand to a Not.

Boolian Algebra - Nomenclature.

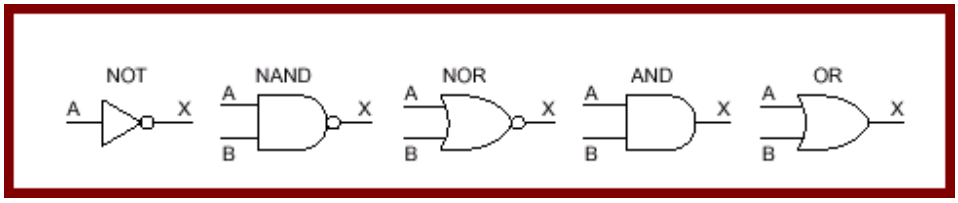
If we designate A and B as inputs to a gate, and X as the output we write:

AB = X for A *And* B = X

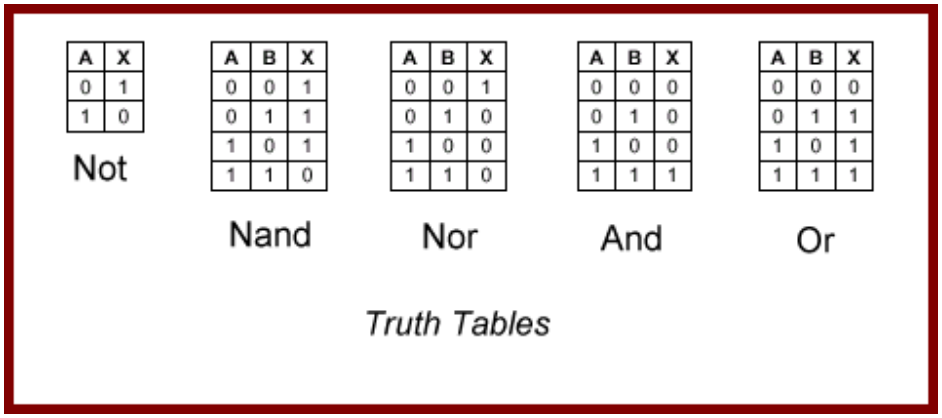
A+B = X for A *Or* B = X

$\overline{AB}$ = X for *Not* (AB) = X

Symbols



Truth Tables



Identities

These are the rules of boolian algebra - similar to the rules of normal alebra. Eg. in normal algebra A x B = B x A, but A - B ≠ B - A.

Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\overline{A} = 0$	$A + \overline{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan's law	$\overline{AB} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A}\,\overline{B}$

Figure 3-6. Some identities of Boolean algebra.

Given these identities, with a bit of thought it can be seen that there is more than one way to accomplish the same truth-table.

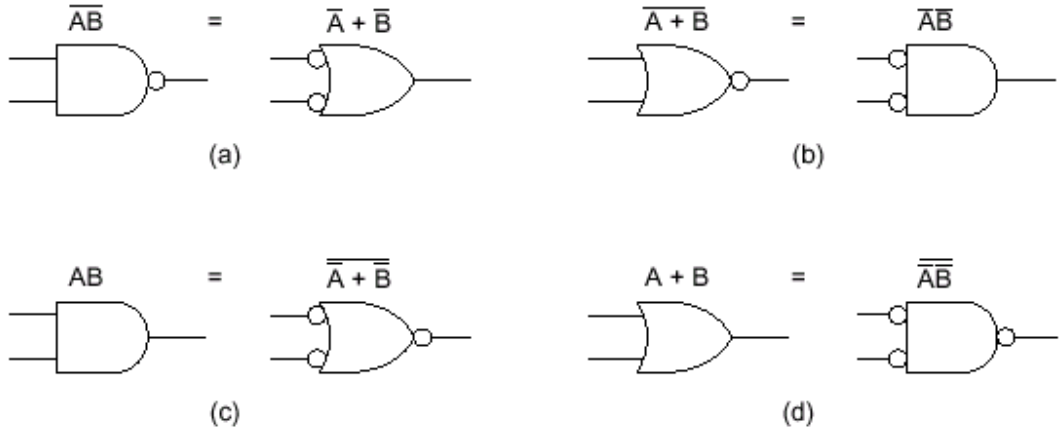


Figure 3-7. Alternative symbols for some gates: (a) NAND. (b) NOR. (c) AND. (d) OR.

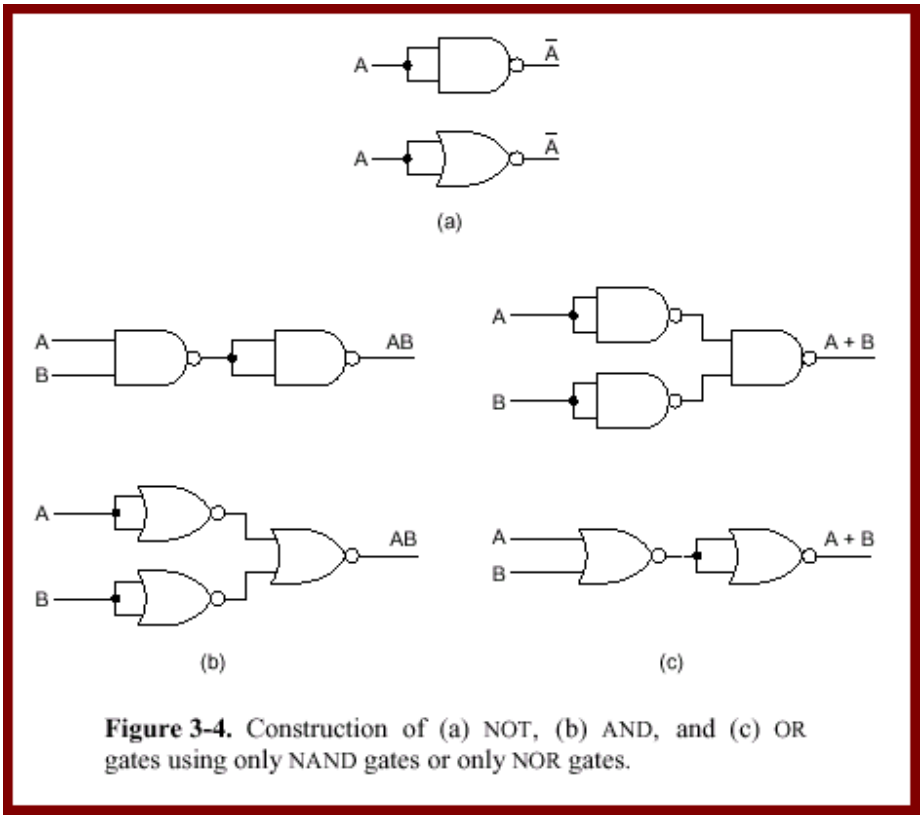
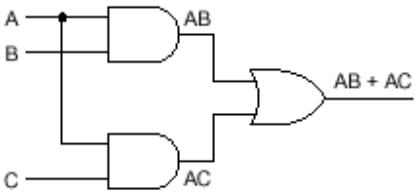
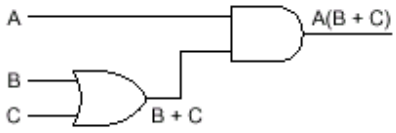


Figure 3-4. Construction of (a) NOT, (b) AND, and (c) OR gates using only NAND gates or only NOR gates.



A	B	C	AB	AC	AB + AC
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

(a)

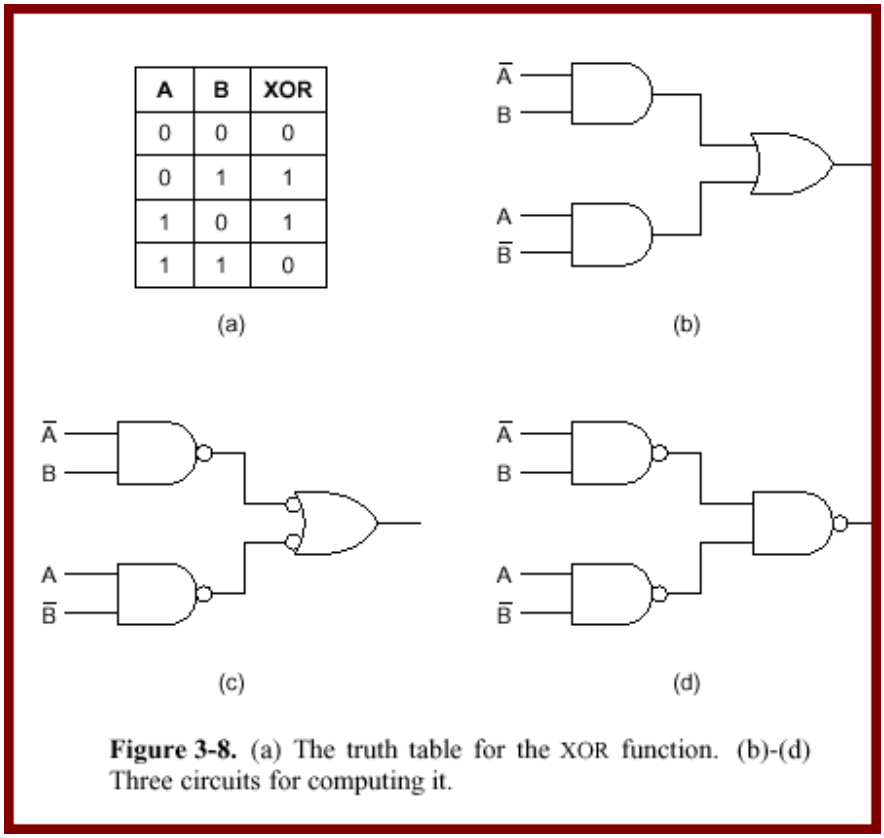


A	B	C	A	B + C	A(B + C)
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1

(b)

Figure 3-5. Two equivalent functions. (a) $AB + AC$. (b) $A(B + C)$.

XOR



A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

(a)

(b)

(c)

(d)

Figure 3-8. (a) The truth table for the XOR function. (b)-(d) Three circuits for computing it.

Basic Digital Logic Circuits

Combinational Circuits

- **Multiplexers - Selects one of n inputs.**

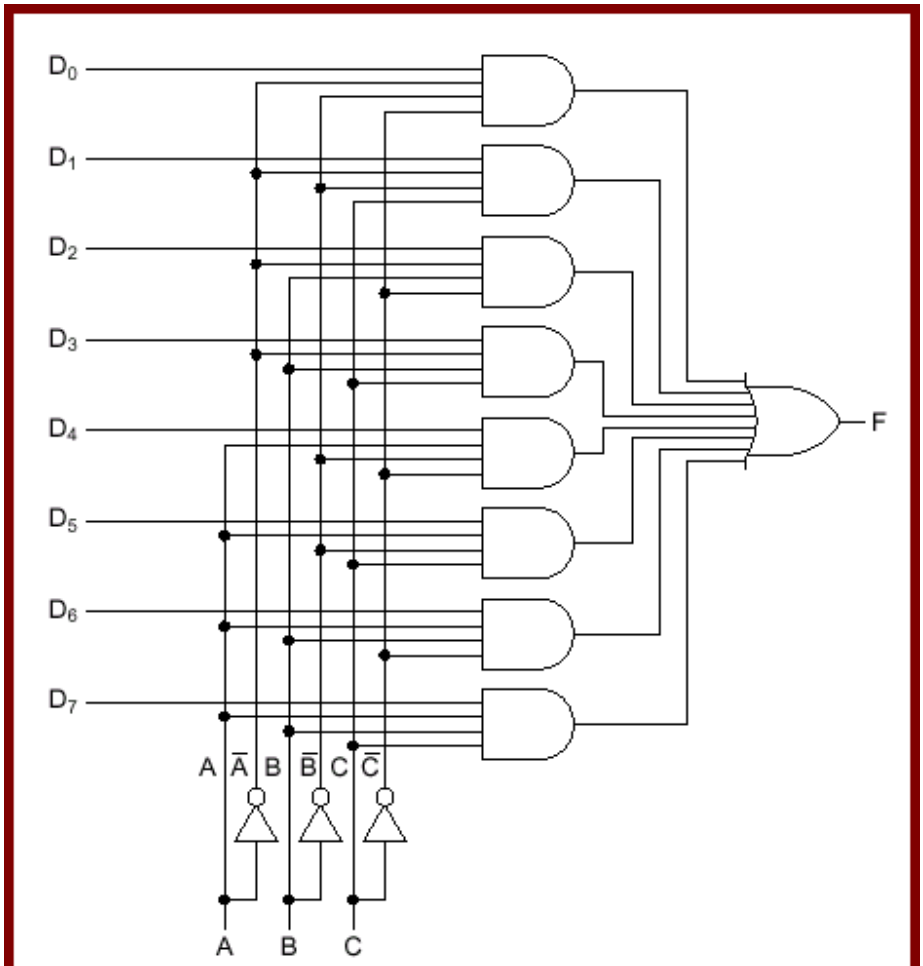


Figure 3-11. An eight-input multiplexer circuit.

- **Decoders - Sets one of n outputs.**

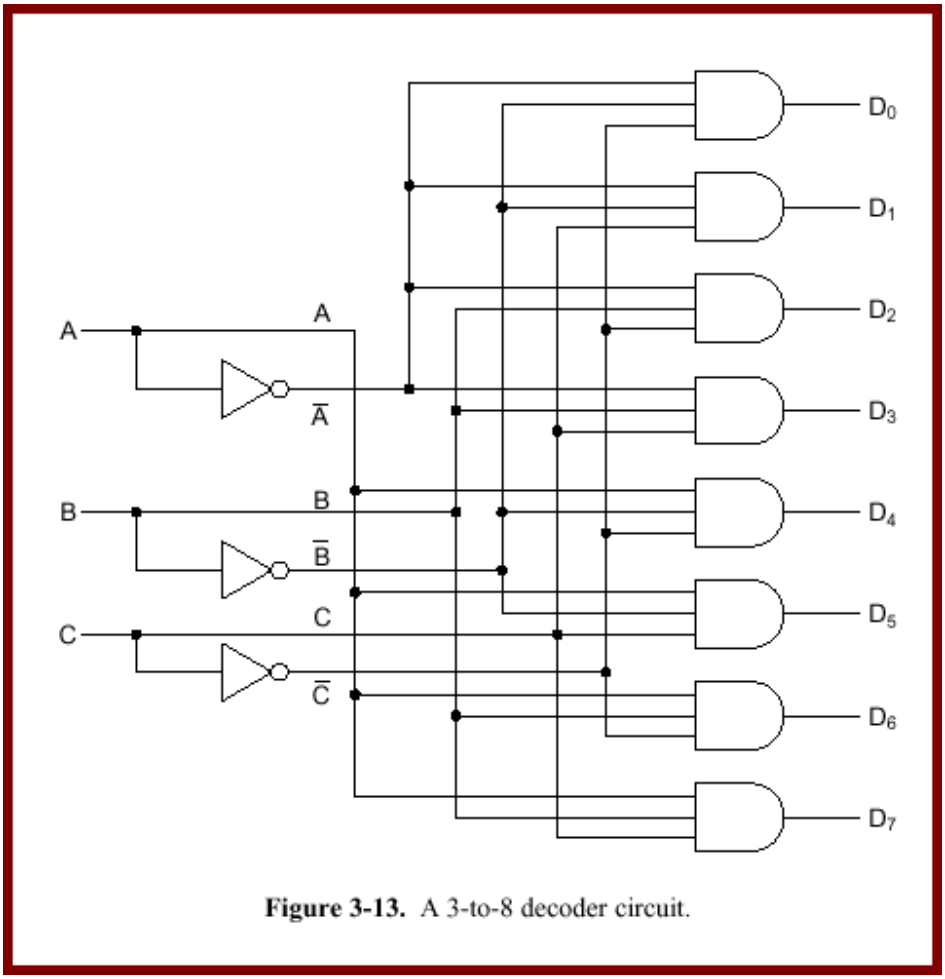


Figure 3-13. A 3-to-8 decoder circuit.

- **Comparators - Are the inputs the same?**

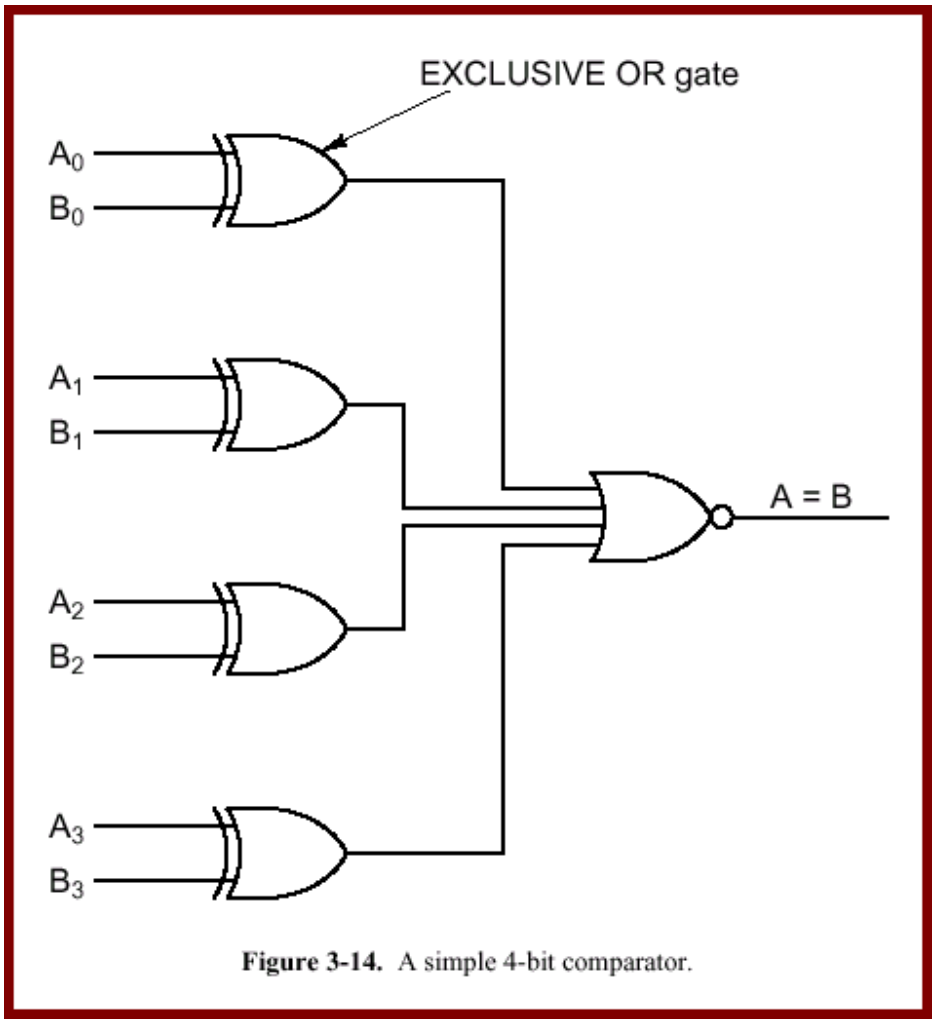


Figure 3-14. A simple 4-bit comparator.

Arithmetic circuits

- **Shifters - Move a bit left or right (multiply/divide by 2).**

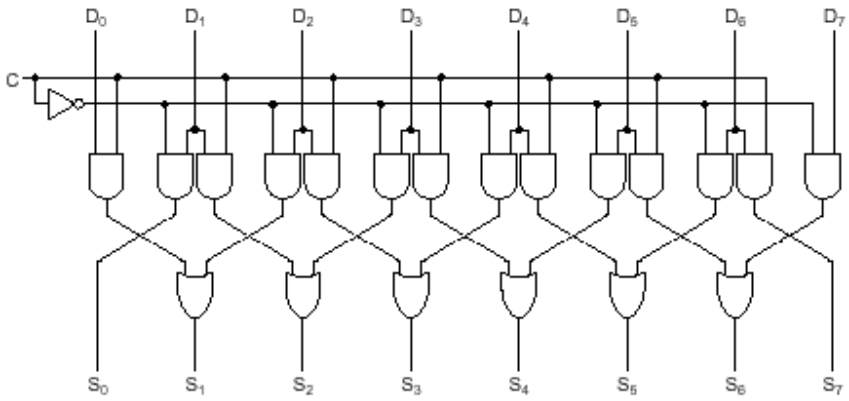


Figure 3-16. A 1-bit left/right shifter.

- **Adders - Add the bits together.**

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

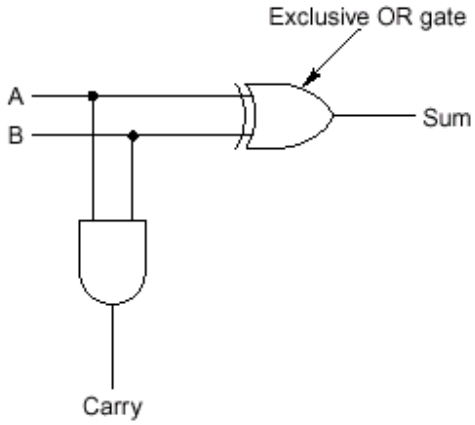
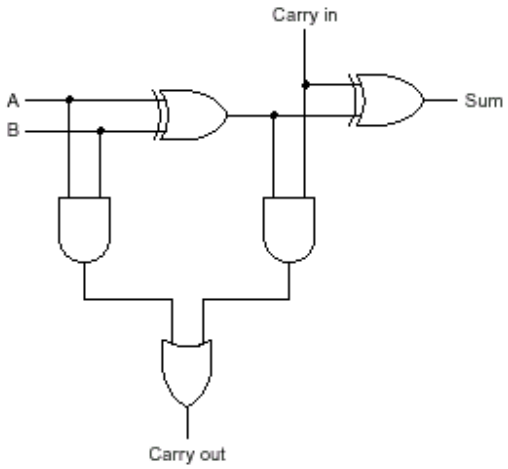


Figure 3-17. (a) Truth table for 1-bit addition. (b) A circuit for a half adder.

A	B	Carry in	Sum	Carry out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

(a)



(b)

Figure 3-18. (a) Truth table for full adder. (b) Circuit for a full adder.

- **ALU's**

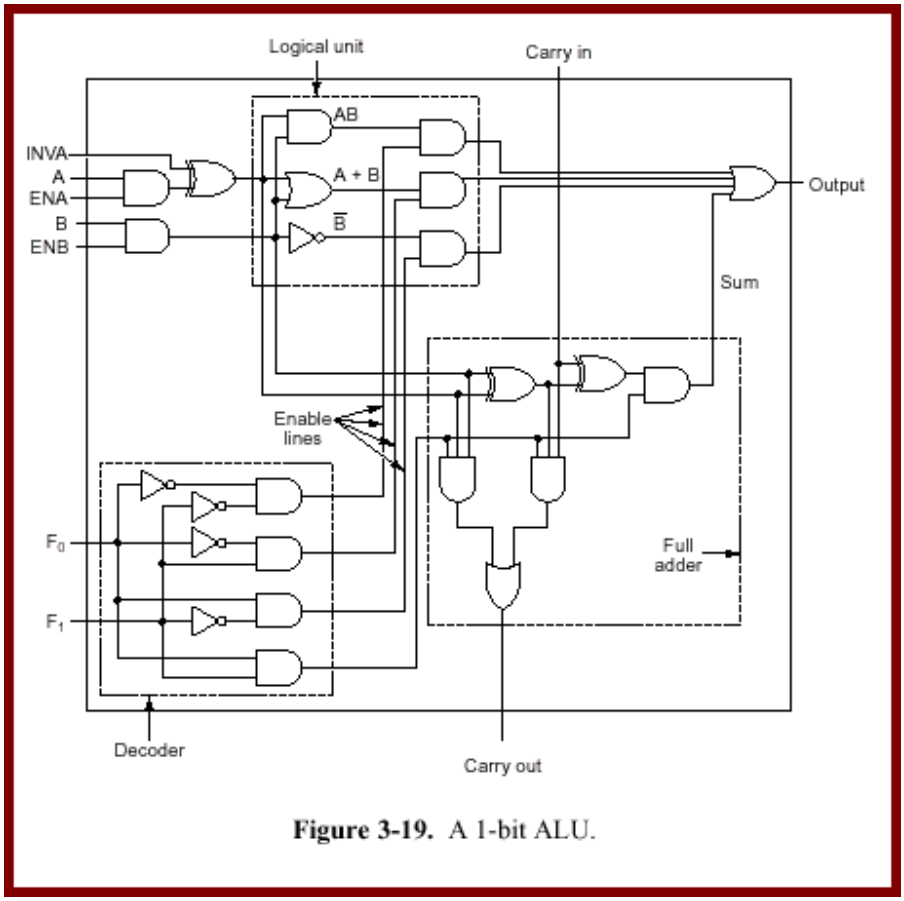


Figure 3-19. A 1-bit ALU.

- **Clocks - When to do it.**

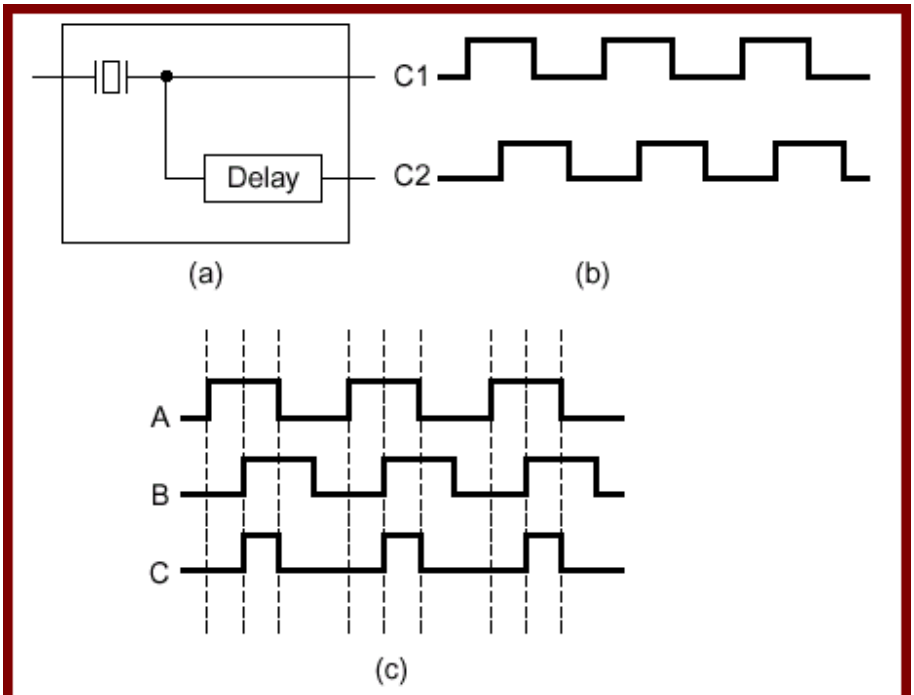
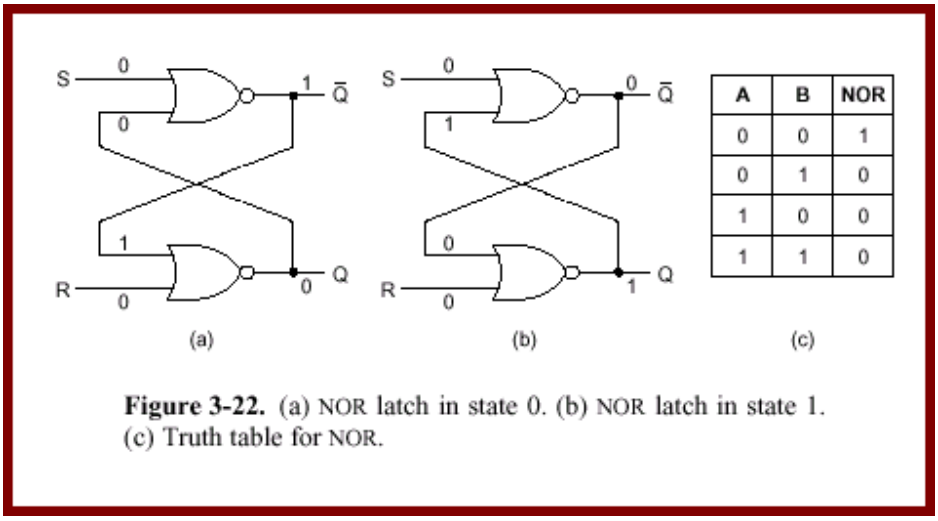


Figure 3-21. (a) A clock. (b) The timing diagram for the clock. (c) Generation of an asymmetric clock.

Memory

- **Latches - switched on input value**



A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

Figure 3-22. (a) NOR latch in state 0. (b) NOR latch in state 1. (c) Truth table for NOR.

- **Clocked Latches set to input value when clocked**

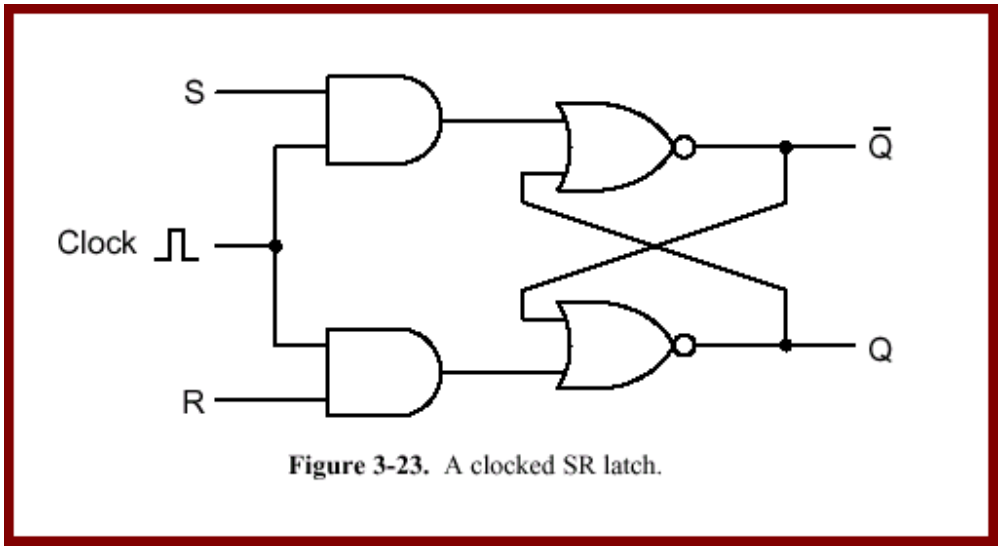


Figure 3-23. A clocked SR latch.

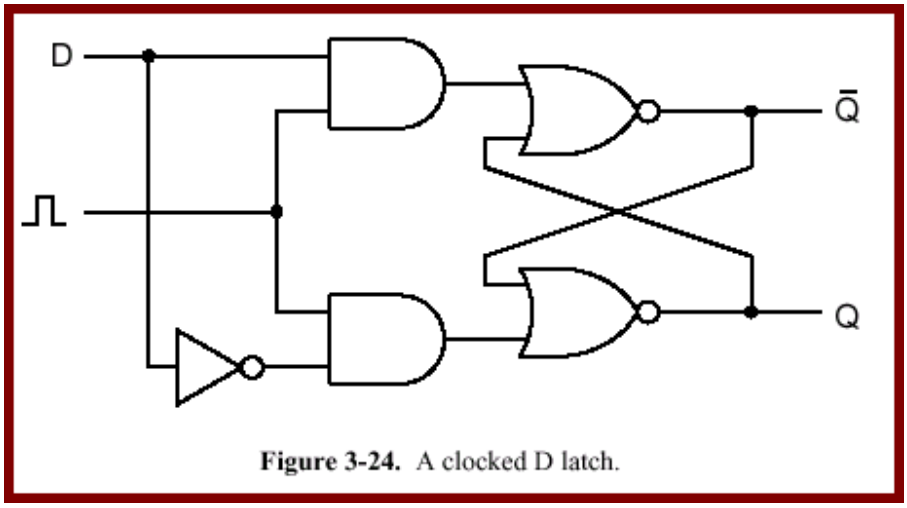


Figure 3-24. A clocked D latch.

- **Rising/falling pulse**

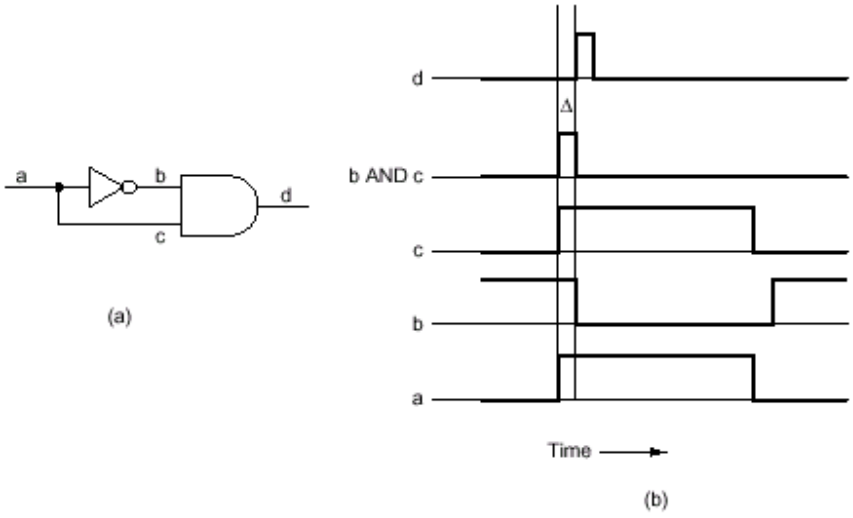


Figure 3-25. (a) A pulse generator. (b) Timing at four points in the circuit.

- **Flip-Flops set on rising or falling pulse**

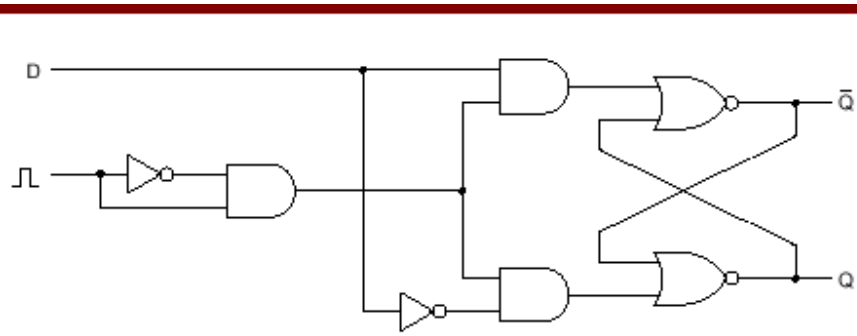


Figure 3-26. A D flip-flop.

- **Registers**

Memory Organization

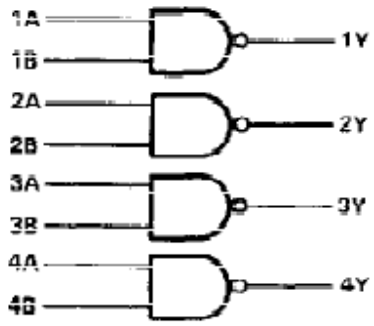
RAM, ROM

Integrated circuits - Some Real *Texas Instruments* Products

[4 Nand gates]

SN5401, SN54LS01,
SN7401, SN74LS01
QUADRUPLE 2-INPUT POSITIVE-NAND GATES WITH OPEN-COLLECTOR OUTPUTS

logic diagram (positive logic)



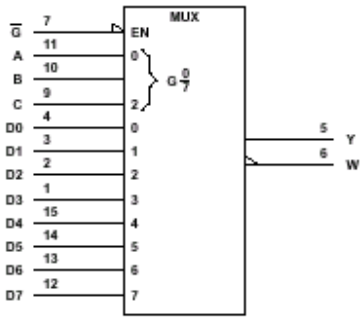
positive logic: $Y = \overline{A \cdot B}$ or $Y = \overline{A} + \overline{B}$

[Multiplexer]

SN54ALS151, SN74ALS151, SN74AS151
1-OF-8 DATA SELECTORS/MULTIPLEXERS

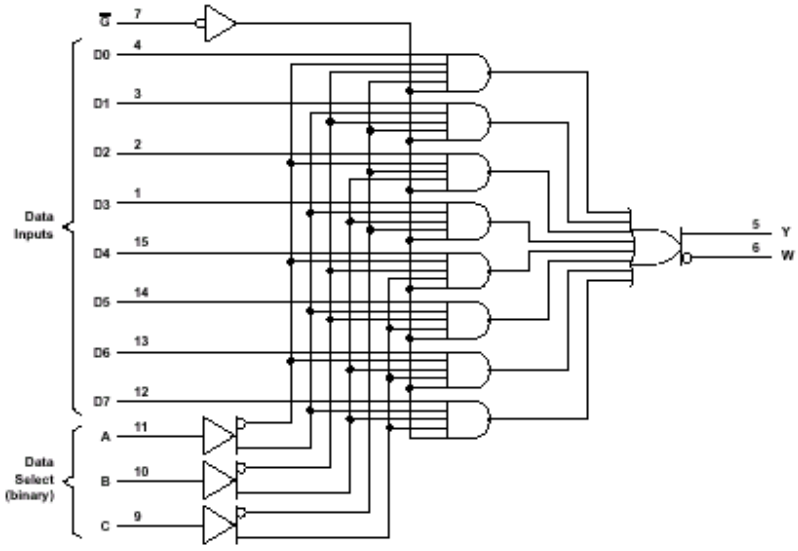
SDAS205A – APRIL 1982 – REVISED DECEMBER 1994

logic symbol†



† This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.
Pin numbers shown are for the D, J, and N packages.

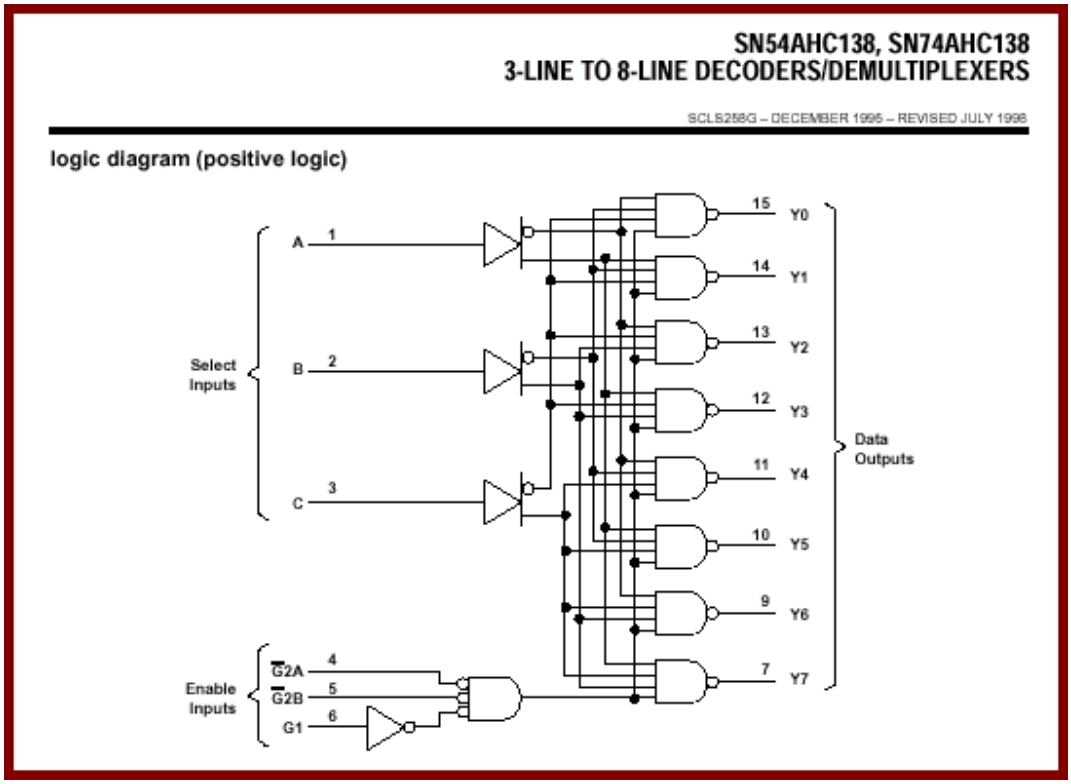
logic diagram (positive logic)



Pin numbers shown are for the D, J, and N packages.

TEXAS INSTRUMENTS

[Decoder]


SN54AHC138, SN74AHC138
3-LINE TO 8-LINE DECODERS/DEMULTIPLEXERS
SCLS258G – DECEMBER 1995 – REVISED JULY 1998
logic diagram (positive logic)
Select Inputs
A
B
C
1
2
3
Enable Inputs
G2A
G2B
G1
4
5
6
15
14
13
12
11
10
9
7
Y0
Y1
Y2
Y3
Y4
Y5
Y6
Y7
Data Outputs

CPU Chips

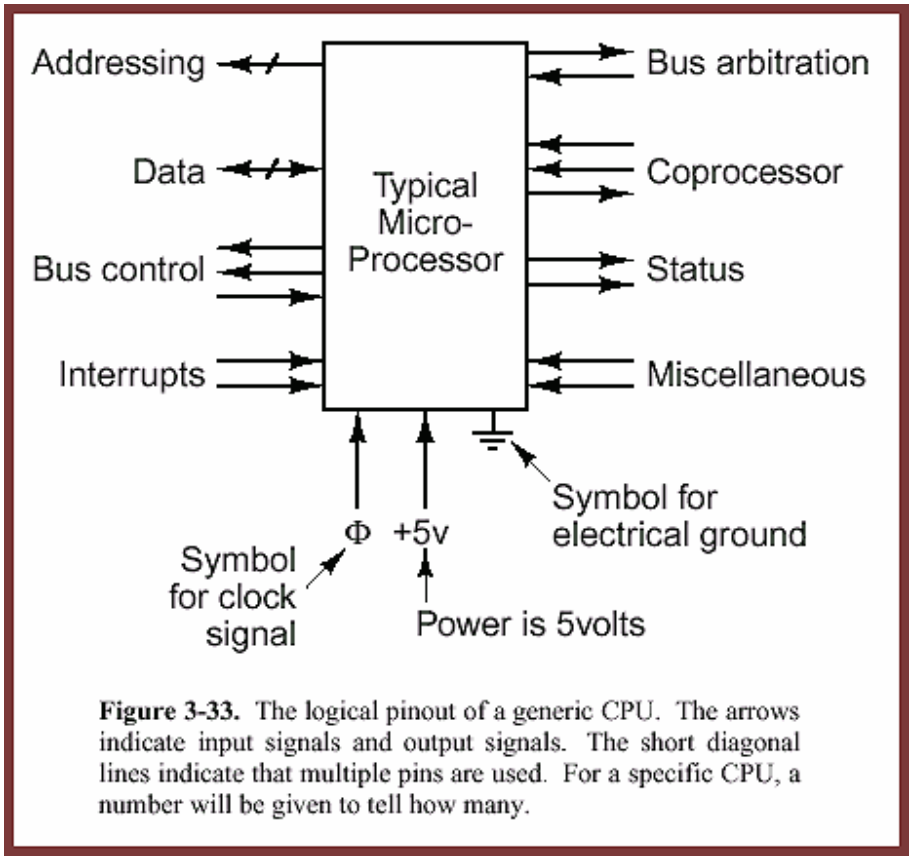


Figure 3-33. The logical pinout of a generic CPU. The arrows indicate input signals and output signals. The short diagonal lines indicate that multiple pins are used. For a specific CPU, a number will be given to tell how many.

Buses

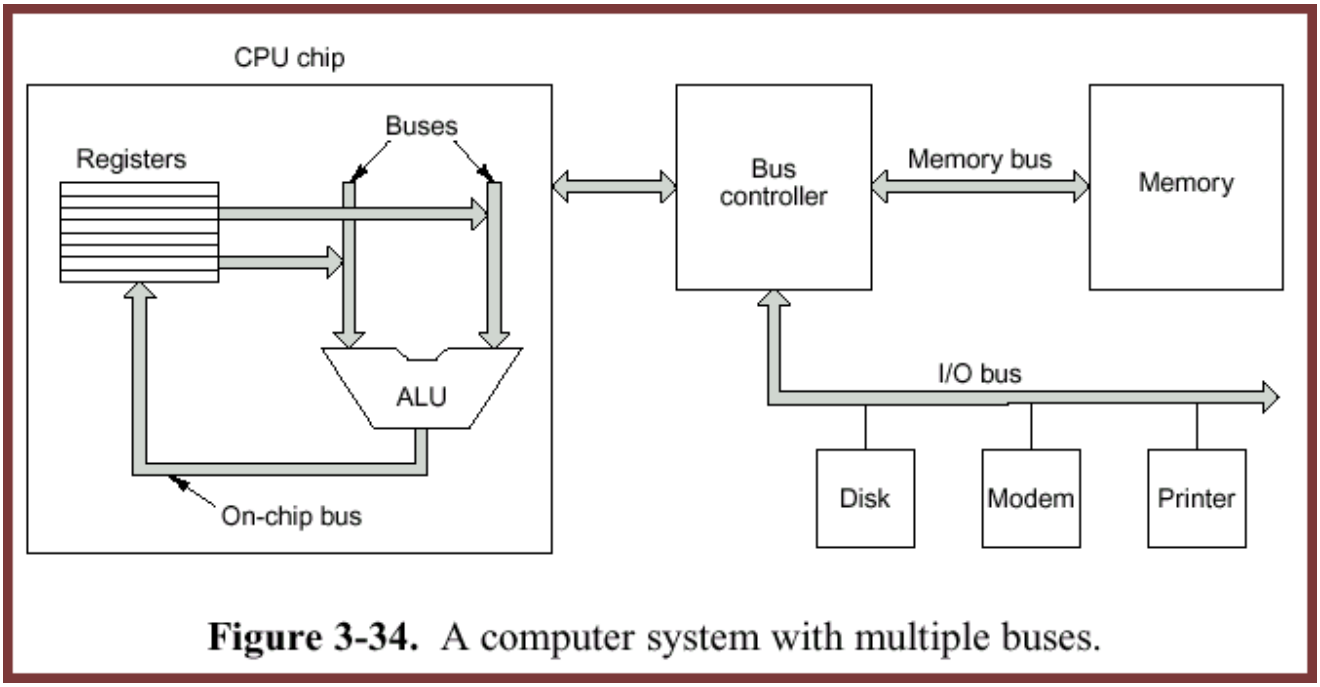


Figure 3-34. A computer system with multiple buses.

Width

Clocking - speed

Syncronous

Asyncronous

Arbitration

Interrupts

Example CPU Chips

- Pentium II
- UltraSPARC II
- picoJava II

Example Buses

ISA

PCI

USB

Interfacing

I/O Chips

PIO - 8255

Address Decoding