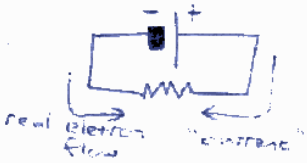
Foundations of Electronics (Digital)

Volts = Joules / Coulomb (work done in moving charge)
Amps = Coulombs / Second (rate of flow of charge)

By convention:



voltage between / across ...
current in / through ...

- current into a point = current out (Kirchoffs current law)
  (thus in a series circuit the current is the same everywhere)
- the sum of the voltage drops around any closed circuit is zero (K's voltage law)

Power = VI ([J/C]·[C/s] = [J/s])

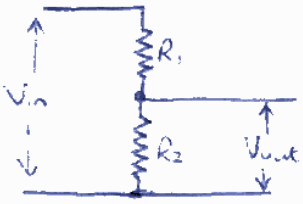
V = IR for <u>conductors</u>

[Shortcut
What is R of:

think of as:

is $\frac{10k\Omega}{3} = 3.33k$]

Voltage Dividers:

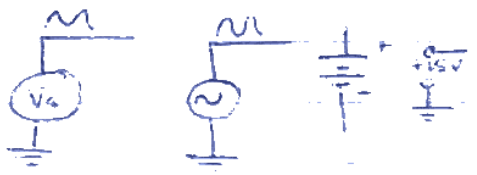


$$V_{out} = IR_2 = \frac{V_{in}}{R_1+R_2} \cdot R_2 = \left(\frac{R_2}{R_1+R_2}\right) V_{in}$$

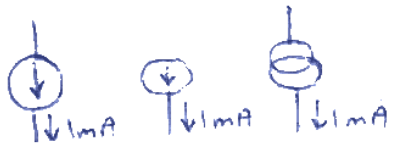
Voltage & Current Sources

~~Notation~~ Symbols:

Voltage Sources

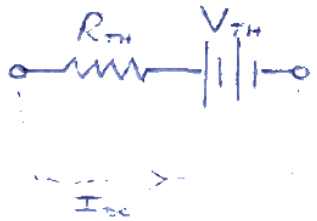


Current Sources



Thevenin's Theorem (& Norton's Theorem)

Any two-terminal network of resistors and voltage sources is equivalent to a single resistor R in series with a single voltage source V (T.T) or a single resistor R in parallel with a single current source I (NT)



V_{TH} is (obviously) the open circuit voltage of the given circuit.

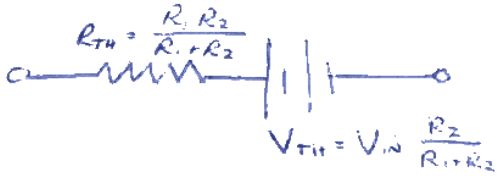
$R_{TH} = V_{TH} / I_{sc}$ where I_{sc} is the short circuit current.

Eg voltage divider:

Open circuit voltage is $V_{IN} \cdot \frac{R_2}{R_1+R_2}$

Short circuit current is V_{IN} / R_1

So Thevenin equiv is



Once we have our Thevenin circuit, and we attach a load, it itself behaves like a voltage divider, and V_{TH} drops (thus a voltage divider is a poor voltage source) — thus we should strive to make $R_{load} \gg R_{internal}$ to minimise "circuit loading". To maximise the power in the

Load $R_{load} = R_{source}$ (however we usually have $R_L >> R_s$)

Small-signal Resistance

For components/devices not subject to ohm's law, we consider the S-S R (incremental resistance, dynamic resistance) $\frac{dV}{di}$

Capacitors

$Q = CV$ ([Farad] = [Coulombs/Volt]

stored charge in cap = capacitance * voltage across cap.

So $I = C \frac{dV}{dt}$

Capacitors cannot dissipate power ~~as~~ even though they conduct, as current & voltage are 90° out of phase.

In parallel $C = C_1 + C_2 + \dots C_n$

In series $C = \frac{1}{\frac{1}{C_1} + \frac{1}{C_2} + \dots + \frac{1}{C_n}}$

DIGITAL ELECTRONICS

EE327F

lectured by : H BRADLOW

from : 24 Feb 1986 – 26 May 1986

DIGITAL ELECTRONICS

2nd year digital electronics

1. DIGITAL CONCEPTS 1
2. A BASIC TRANSISTOR MODEL 2
3. NUMBER CODING SYSTEMS 5
4. BASIC LOGIC GATES 11
5. BOOLEAN ALGEBRA 13
6. KARNAUGH MAPS 15
7. BASIC COMBINATIONAL LOGIC 18
8. BI-STABLES & FLIP-FLOPS 25
9. SEQUENTIAL DIGITAL CIRCUITS 31

3rd year d.e.

10. IMPLEMENTATION OF NUMBER CODING SYSTEMS 35
11. SEQUENTIAL & COMBINATIONAL LOGIC CIRCUITS. 38
12. DIGITAL TECHNOLOGIES 59
13. APPLICATION SPECIFIC INTEGRATED CIRCUITS 96
14. A/D & D/A CONVERTORS 105

1 DIGITAL CONCEPTS

1.1 WHY DIGITAL?

- accuracy within discrete intervals
- inherently digital nature of calculation
- ability to store and process data
- 'stored program control'

1.2 WHY BINARY?

- ease of implementation (switching technology)
- Boolean operations
- noise rejection and drift less of a problem as we work with voltage ranges. Binary allows the largest range size and separation.

CONVERTING DECIMAL TO BINARY

INTEGER PART: Divide successively by two, writing down the remainders. In reverse order, these constitute the binary number.

FRACTIONAL PART: Multiply successively by two, noting the integral carry each time, and then taking only the fractional part

		CARRY
eg	$0.85 \times 2 = 0.7$	1
	$0.7 \times 2 = 0.4$	1
	$0.4 \times 2 = 0.8$	0
	$0.8 \times 2 = 0.6$	1
	$0.6 \times 2 = 0.2$	1
	$0.2 \times 2 = 0.4$	0

So 0.85 is 0.110110 to 6 places or $0.1101100110\overline{0011}$

2. A Basic Transistor Model.

The transistor is an "active" component, a device which can amplify, producing an output signal with more power in it than the input signal. The additional power comes from an external source, the power supply.

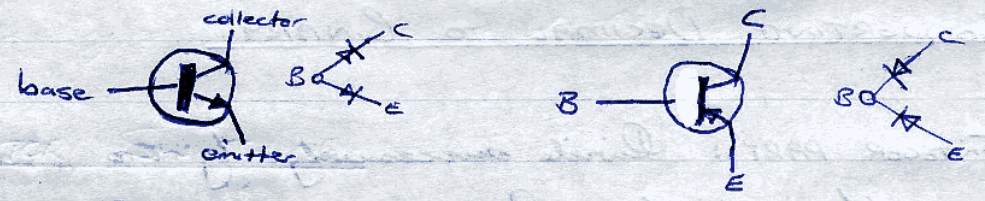
Notation: • Voltage at a single terminal relative to ground is V_C, V_E and V_B.

• Voltage between two terminals has a double subscript. eg V_{BE} is the base-to-emitter voltage drop

• The same subscript twice a power supply voltage, V_{CC} being +ve and V_{EE} negative (npn)

2.1 The Basic Current Amplifier Model.

Two types: npn pnp



Rules for NPN (for pnp reverse all polarities)

1 • collector must be more positive than emitter

2 • base-emitter circuit behaves like a conducting diode.

• base-collector circuit behaves like a reversed-bias diode.

3 • any given transistor has maximum values for I_C, I_B, V_{CE}, power dissipation $I_C V_{CE}$, temperature, etc, which cannot be exceeded without damaging the t.

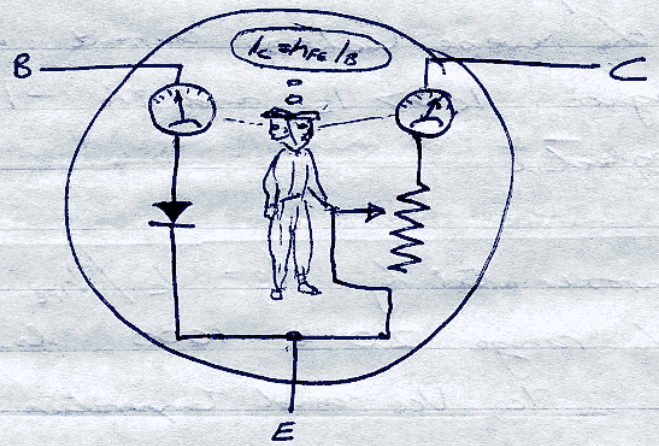
When rules 1-3 apply

4 • I_C is roughly proportional to I_B, ie $I_C \simeq h_{FE} I_B$ where h_{FE}, the current gain (also called β) is typically 100. Warning: h_{FE} is not a reliable constant

and a circuit that depends on a particular value of h_{FE} is a bad circuit

Property 2 shows that you cant apply a voltage across the base emitter terminals as an enormous current will flow if the base is more positive than the emitter by more than about 0.6 to 0.8 Volts (forward diode drop) This implies $V_B \approx V_E + 0.6$ V (ie $V_B = V_E + V_{BE}$) (again npn polarities)

An alternative model:



The man's task is to try to keep $I_C = h_{FE} I_B$; however, he is only allowed to vary resistance. He can thus go from a short circuit (saturation) to an open circuit (transistor "off") or anything in between, but he isn't allowed to use batteries, current sources, etc.

At any given time, a transistor may be:

- cut off (no collector current)
- active ($V_C > V_E + V_X$, where $V_X \approx 0.2$-0.4 V)
- saturated ($V_C \approx V_E$)

2.2 THE EBERS - MOLL MODEL.

For some applications, the previous model is inadequate. We need to think of a transistor as a transconductance amplifier : collector <u>current</u> is determined by base-to emitter <u>voltage</u>. The modified property 4 becomes:

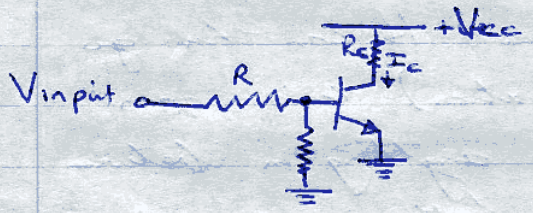
$$I_c = I_s \left[e^{\frac{V_{BE}}{V_T}} - 1 \right]$$

where : $V_T = kT/q = 25.3$ mV at room temperature

I_s is the reverse leakage current : in the active region $I_c \gg I_s$ and the -1 term can be neglected.

I_B can be approximated by $I_B = I_c / h_{FE}$ where h_{FE} is typically in the range 20 to 1000, but depends on transistor type, I_c, V_{CE} and temperature

TRANSISTOR SWITCHING



Obviously $I_c(max) = \frac{V_{cc}}{R_c}$ (saturation current)

(gnd) no input : $I_c = 0$ (V_{input} gets tied to ground)

input : V_{BE} rises to 0,6 V (forward diode drop)

Drop across R_{in} is : $V_{in} - 0{,}6$ V

$\Rightarrow$ Base current $= (V_{in} - 0{,}6V)/R_{in}$

As saturation current is $\frac{V_{cc}}{R_c}$, saturation occurs for $I_B \geq \frac{V_{cc}}{\beta R_c}$ where β is typically 100.

3 NUMBER CODING SYSTEMS

3.1 BCD (usually 8421. 4221 also common, as well as XS-3)

We use a 4-bit group for each decimal digit. Commonly used for I/O of decimal numeric data (eg signals to calculator display drivers, etc). Conversion to binary is complicated

3.2 SIGN-MAGNITUDE

Simple method: devote 1 bit (the MSB usually) to the sign, with the remaining bits representing the magnitude. Used when numbers are displayed as well as in some A/D conversion schemes.
Subtraction is difficult (choose sign from larger number, magnitude from the difference between the larger and smaller magnitudes)
and there are two zeroes.

3.3 OFFSET BINARY

Subtract half the largest possible number to get the value represented. The MSB still carries the sign information and there is only one. Popular in D/A, A/D applications but still difficult for computations.

3.4 ONE'S COMPLEMENT.

Must specify word length in complement notations
Range (1-4bit) 0 : 0/0000 −0 : 1/1111
to +15 : 0/1111 to −15 : 1/0000

Addition : add
Subtraction of B from A : Take 1's C of B, add to A, and add the carry bit

3.5 2's - COMPLEMENT

A negative number is the binary number you add to a positive value of the same magnitude to get a zero result. Only one convenient zero 0000..0

Arithmetic is simple: addition - binary add (ignore carry!)
subtract B from A: take 2's-C of B and add.
negate: take 2's complement.

— 4-BIT NUMBER REPRESENTATIONS —

INTEGER	SIGN MAGNITUDE	OFFSET BINARY	ONE'S COMP	TWO'S COMP
7	0111	1111	0111	0111
6	0110	1110	0110	0110
5	0101	1101	0101	0101
4	0100	1100	0100	0100
3	0011	1011	0011	0011
2	0010	1010	0010	0010
1	0001	1001	0001	0001
0	0000	1000	0000	0000
-1	1001	0111	1010	1111
-2	1010	0110	1101	1110
-3	1011	0101	1100	1101
-4	1100	0100	1011	1100
-5	1101	0011	1010	1011
-6	1110	0010	1001	1010
-7	1111	0001	*1000	1001
-8	—	0000	—	*1000
-(0)	1000	—	1111	—
	—	—	-7	-8

weight of sign bit

WEIGHTING OF THE SIGN BIT. (COMPLEMENT NOTATION).

We consider the binary number for which the only non-zero bit is the sign bit.
In general for a $n+1$ bits wordlength, the weight of the sign bit is $-(2^n-1)$ in 1's complement, and -2^n in 2's complement.

MULTIPLICATION AND DIVISION

eg:

```
  7        111
 x5       x101
 --       ----
 35        111
        +11100
        ------
        100011
```

```
        6
2 )12        10 )1100
                 110
                 ---
                  10
                  10
                  10
                  ---
                  000
```

3.6 EXCESS-3 BCD.

The XS-3 code for a decimal number is obtained by converting to binary and adding 3. The forbidden codes are (decimal) 0, 1, 2, 13, 14, 15

Addition. Begin by adding the least significant digits.
If there is no carry, the resulting digit is XS-6 and we must subtract 3.
If there is a carry, it has weight 16, so the result is XS-0 (10 for decimal carry, plus 6) and we add 3.
Continue, taking the carry bit across digits to MSD.

eg

09		0011	1100	XS-3
06		0011	1001	XS-3
15	XS-6 →	0111	0101	← XS-0 as it results in a carry.
		−3	+3	
		0100	1000	XS-3

ASIDE ON COMPLEMENTS:

A number and its 1's complement will add up to 1111
" " " " 2's complement will add up to 2222
⋮
" " " " 9's " " " " " 9999
" " " " 10's " " " " " 0009

The 2's complement of a binary number added to the number is equivalent to 2 in each column

eg

3	0011
−3	1101
	0000

equivalent to a 2 in each column.

whereas 1's complement will add up to 1111.

9's and 10's complement are the decimal equivalent.

eg

716		716
283	(9's C)	394
999		0000

XS-3 is self-complementing (see opposite)
ie 1's C of XS-3 gives 9's C of decimal.

Decimal Digit	XS-3	1's C of XS-3	Decimal of 1's C.
0	0011	1100	9
1	0100	1011	8
2	0101	1010	7
3	0110	1001	6
4	0111	1000	5
5	1000	0111	4
6	1001	0110	3
7	1010	0101	2
8	1011	0100	1
9	1100	0011	0

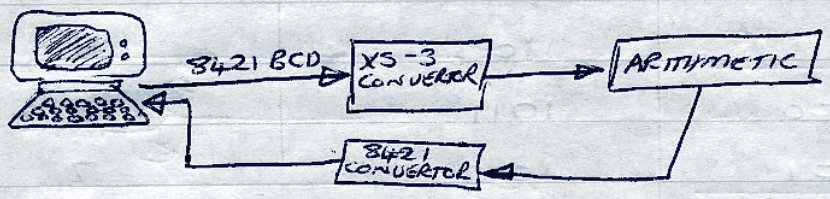
If we have XS-6 we can add 1101 (equivalent to subtracting 3 in 2's-C). We need not use the sign bit as this will always be zero (this because $1101_2 = 13_{10}$, and we are adding to at least 6, ie a bit will be carried, which together with the sign bit (1) causes a sign bit of zero)

eg

09	0011	1100
08	0011	1011
17	0111	0111
	+ 1101	0011
	0100	1010

Ie, if carry add 0011 to the digit <u>causing</u> carry
else add 1101 (ie add $\overline{C}\overline{C}C1$ of XS-3 BCD addition diagram)

Typical uses:

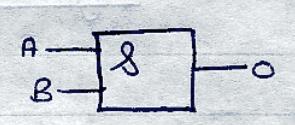


4 Basic Logic Gates.

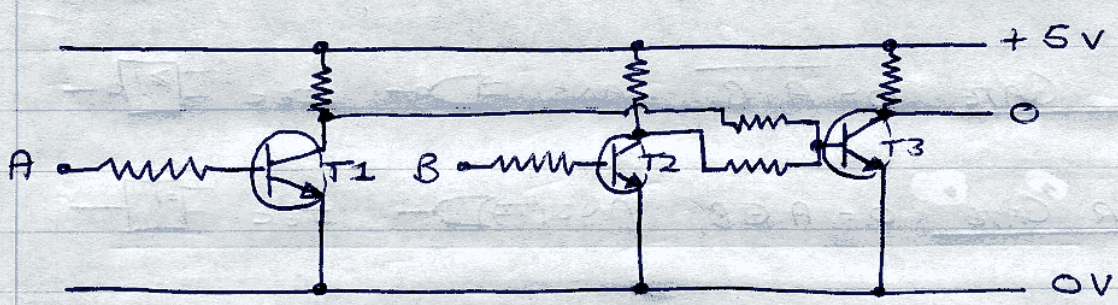
4.1 And Gate

ANSI

IEC



$O = A \cdot B$



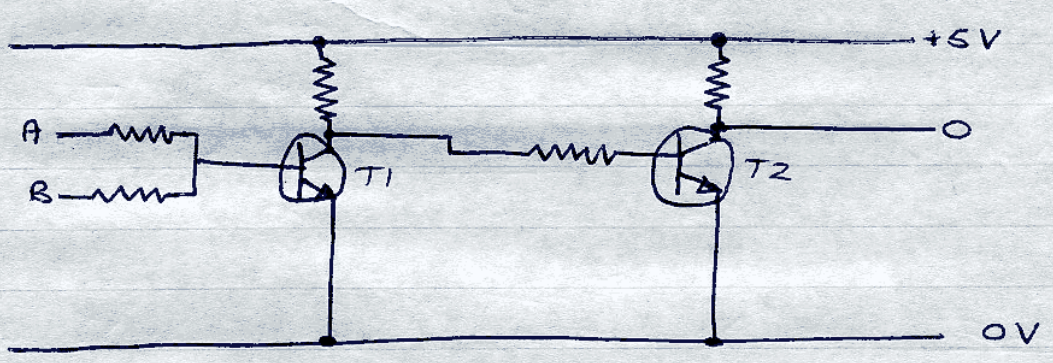
A	B	T1	T2	T3	O
Low	Low	Off	Off	On	Low
Low	High	Off	On	On	Low
High	Low	On	Off	On	Low
High	High	On	On	Off	High.

4.2 OR Gate

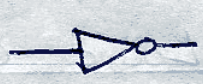
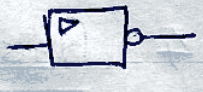
ANSI

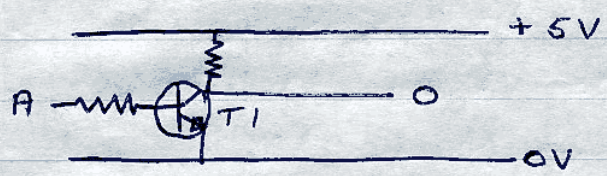
IEC

$O = A + B$



A	B	T1	T2	O
Low	Low	Off	On	
Low	High	On	Off	
High	Low	On	Off	
High	High	On	Off	

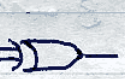
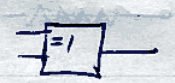
4.3 NOT GATE. $O = \overline{A}$ ANSI  IEC 



A	T1	O
Low	Off	High
High	On	Low

4.4 NAND GATE. $O = \overline{A \cdot B}$ ANSI IEC

4.5 NOR GATE $O = \overline{A + B}$ ANSI IEC

4.6 EXOR GATE $O = A \oplus B$ ANSI  IEC 

5 Boolean Algebra.

5.1 Properties:

Commutative $A \cdot B = B \cdot A$ $\quad A + B = B + A$

Distributive $A \cdot (B + C) = A \cdot B + A \cdot C$ $\quad A + (B \cdot C) = (A + B) \cdot (A + C)$

Associative $A \cdot (B \cdot C) = (A \cdot B) \cdot C = (A \cdot C) \cdot B$

$A + (B + C) = (A + B) + C = (A + C) + B.$

5.2 Identities:

$A \cdot 1 = A$ $\quad A + 1 = 1$ $\quad \overline{\overline{A}} = A$

$A \cdot 0 = 0$ $\quad A + 0 = A$

$A \cdot A = A$ $\quad A + A = A$

$A \cdot \overline{A} = 0$ $\quad A + \overline{A} = 1$

5.3 De Morgan's Laws. (Proof trivial by truth tables).

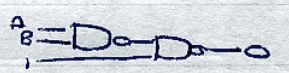
$\overline{A + B} = \overline{A} \cdot \overline{B}$ or $A + B = \overline{\overline{A} \cdot \overline{B}}$

$\overline{A \cdot B} = \overline{A} + \overline{B}$ or $A \cdot B = \overline{\overline{A} + \overline{B}}$

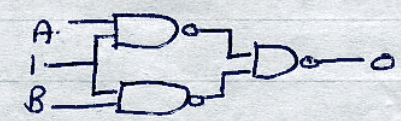
5.4 Making all the Basic Logic Functions from NAND.

NOT: $0 = \overline{A \cdot 1}$

AND $0 = \overline{\overline{A \cdot B}} = \overline{\overline{A \cdot B} \cdot 1}$



OR $0 = \overline{\overline{A} \cdot \overline{B}} = \overline{\overline{A \cdot 1} \cdot \overline{B \cdot 1}}$



6 KARNAUGH MAPS.

The basic technique used in the simplification of a Boolean expression is to combine two terms that differ only in one variable that changes to its complement, into one

eg $A \cdot B \cdot C + A \cdot B \cdot \overline{C} = A \cdot B$.

Karnaugh maps provide an easy way to detect such groupings. A KM is a graphic representation of a truth table, with a square provided for each possible input combination (ie conjunction), so there are 2^n squares, where n is the number of input variables. The squares are arranged as a rectangular matrix.

eg: 2-inputs

	B	$\overline{B}$
A	$A \cdot B$	$A \cdot \overline{B}$
$\overline{A}$	$\overline{A} \cdot B$	$\overline{A} \cdot \overline{B}$

3 inputs

	$B \cdot C$	$B \cdot \overline{C}$	$\overline{B} \cdot \overline{C}$	$\overline{B} \cdot \overline{C}$
A				
$\overline{A}$				

Once we have constructed the map we insert the functional truth value of each square.

<u>NOTE</u> Any two adjacent rows/cols on the Karnaugh map have only one input different, and these are complements.

We obey the following grouping rules

- all 1's must be grouped
- 1's may be grouped more than once in overlapping groups.
- group sizes are powers of 2.

The simplified value of the function is the OR function of the values of the groupings. The value of a group is calculated by looking at the variables that do not change within the group. Groupings may wrap around the end of the matrix.

The procedure is thus as follows:

- Draw a Karnaugh map for a function of n variables
- Enter the fundamental products into the map
- Group all possible octets (8×1 or 4×2)*
- Group all possible quads (4×1 or 2×2)
- Group all possible pairs
- Group all possible singles
- Read off the function from the group values.

* <u>NOTE:</u> An n-variable problem produces a K.M. of size 2^n. The grouping sizes are thus $2^n, 2^{n-1}, \ldots, 1$. If a 2^n group exists, the function is TRUE. Eg a 3-variable problem with an octet group necessarily implies TRUE for all cases.

<u>EXAMPLE:</u> Simplify $F = \bar{A}.B.\bar{C} + A.\bar{B}.\bar{C} + A.\bar{B}.C + A.B.\bar{C} + A.B.C$

	$\bar{B}.\bar{C}$	$B.\bar{C}$	$B.C$	$\bar{B}.C$
A	1	1	1	1
$\bar{A}$		1		

Quad has value A. $(2^{3-1})^{\dagger}$

Pair has value $B.\bar{C}$ $(2^{3-2})^{\dagger}$

Simplified function: $F = A + B.\bar{C}$

<u>NOTE:</u> The inverse function can be read off by grouping the zeroes.

† Furthermore, a 2^{n-1} group is equivalent to a single variable, and in general a 2^{n-j} group is equivalent to a conjunction of j variables

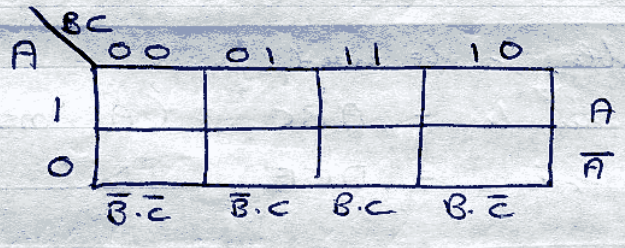
GREY CODE.

If the K.M. is being constructed directly from a truth table it is sensible to label the axes with a binary code. The normal sequence 00, 01, 10, 11.. will not work, as for example, in the transition from 01 to 10 both states change. We need a binary sequence in which only one bit changes during the transmission from one number to the next – a Grey Code

Decimal	Binary	Gray Code
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

A	B	C

Eg 3-variable Karnaugh Map



Advantages: avoids 'glitches' due to racing as only one input changes at a time.

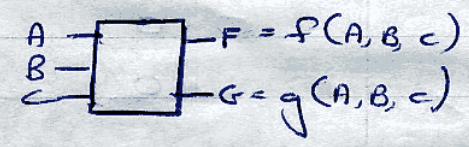
7 BASIC COMBINATIONAL LOGIC

A digital circuit can be thought of as a block with a number of inputs & outputs, all of which are logic values.

In a <u>combinational logic circuit</u> the output(s) are simply logic functions of the input variables. (In a sequential circuit, the output depends also on the state of the circuit; see later).

COMBINATIONAL CIRCUITS WITH MORE THAN ONE OUTPUT.

Each output is represented by a boolean function of the input variables. eg:



• CODE CONVERSION CIRCUITS.

<u>Example 1</u>: design a circuit to convert a 3-bit natural binary number into 3-bit grey code.

<u>Step 1</u>: define variables to be used
- binary number = ABC (A is MSB)
- grey code = DEF

<u>Step 2</u>: define function in the form of a truth table

Decimal	A	B	C	D	E	F
0	0	0	0	0	0	0
1	0	0	1	0	0	1
2	0	1	0	0	1	1
3	0	1	1	0	1	0
4	1	0	0	1	1	0
5	1	0	1	1	1	1
6	1	1	0	1	0	1
7	1	1	1	1	0	0

Step 3: Convert the function into Boolean expressions if necessary.

$$D = A \cdot \bar{B} \cdot \bar{C} + A \cdot \bar{B} \cdot C + A \cdot B \cdot \bar{C} + A \cdot B \cdot C$$

$$E = \bar{A} \cdot B \cdot \bar{C} + \bar{A} \cdot B \cdot C + A \cdot \bar{B} \cdot \bar{C} + A \cdot \bar{B} \cdot C$$

$$F = \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot C + A \cdot B \cdot \bar{C}$$

Step 4: Simplify expressions

	$\bar{B}\cdot\bar{C}$	$\bar{B}\cdot C$	$B\cdot C$	$\bar{B}\cdot C$
$\bar{A}$				
A	1	1	1	1

$D = A$

	$\bar{B}\cdot\bar{C}$	$\bar{B}\cdot C$	$B\cdot C$	$\bar{B}\cdot C$
$\bar{A}$			1	1
A	1	1		

$E = A \cdot \bar{B} + \bar{A} \cdot B = A \oplus B$

	$\bar{B}\cdot\bar{C}$	$\bar{B}\cdot C$	$B\cdot C$	$\bar{B}\cdot C$
$\bar{A}$		1		1
A		1		1

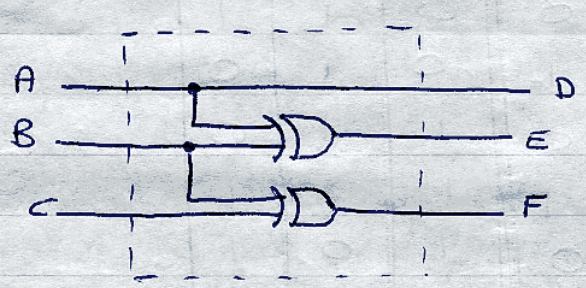
$F = \bar{B} \cdot C + B \cdot \bar{C} = B \oplus C$

Step 5: Draw the simplified circuit

$D = A$

$E = A \oplus B$

$F = B \oplus C$



Example 2 Opposite conversion.

S1: Grey code ABC Binary DEF

S2:

Grey Code			Binary		
A	B	C	D	E	F
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	1
1	0	1	1	1	0
1	1	0	1	0	0
1	1	1	1	0	1

S3:

S4:

$D = A$

A \ BC	00	01	11	10
0 $\bar{A}$	0	0	0	0
1 A	1	1	1	1

	00	01	11	10
0 $\bar{A}$	0	0	1	1
1 A	1	1	0	0

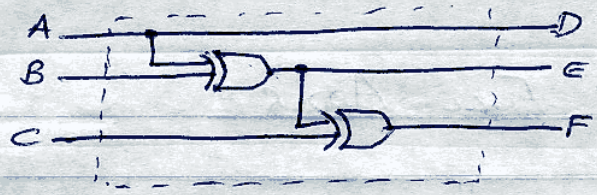
$E = \bar{A} \cdot B + A \cdot \bar{B} = A \oplus B$

A \ BC	00	01	11	10
0 $\bar{A}$	0	1	0	1
1 A	1	0	1	0

$F = \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot \bar{C} + A \cdot B \cdot C$

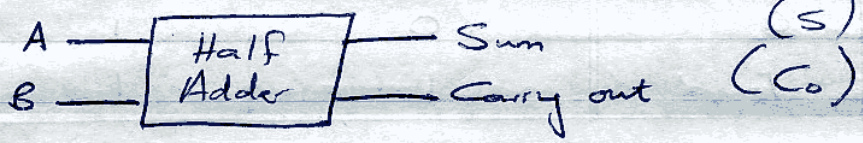
$= A \oplus B \oplus C$

SS:



• ARITHMETIC CIRCUITS

The simplest arithmetic function is the Half Adder (HA):



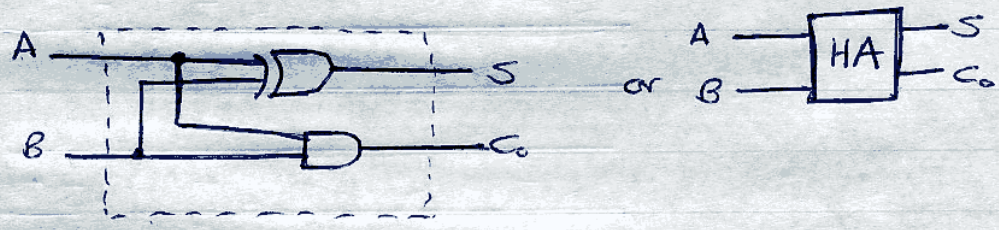
TRUTH TABLE

A	B	S Sum	Co Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

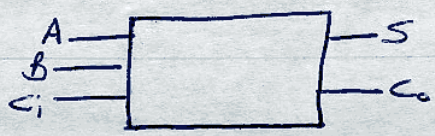
ie $S = \bar{A} \cdot B + A \cdot \bar{B} = A \oplus B$

$C_o = A \cdot B$

Schematic:



This works fine if we're just adding two bits, but if we are using multiple-bit numbers each two bit adder must take into account the carry of the adder on its right. The circuit to do this is a Full Adder

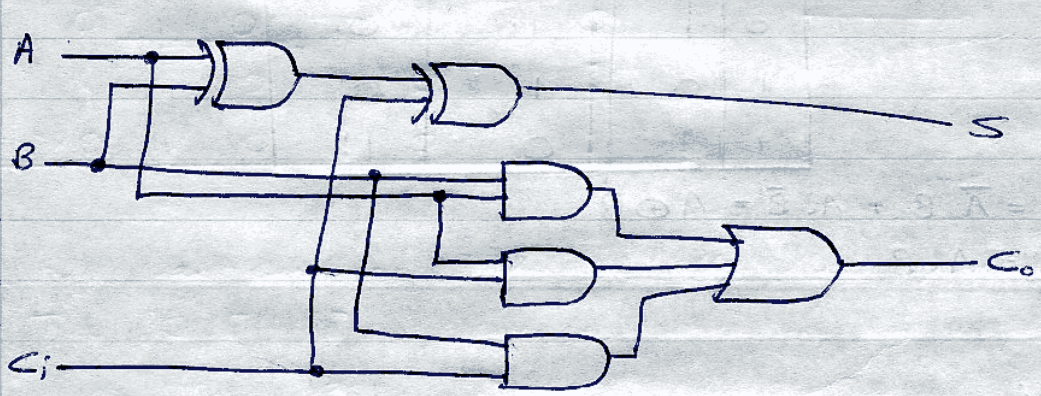


Truth Table of Full Adder

A	B	C_i	S	C_o
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

$$S = A \oplus B \oplus C_i$$

$$C_o = B \cdot C_i + A \cdot C_i + A \cdot B$$

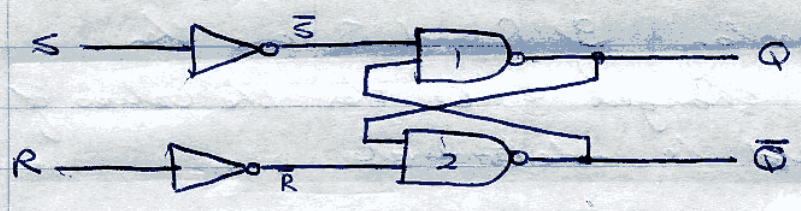


8 BI-STABLES & FLIP-FLOPS.

We now examine some sequential circuits.

R/S (RESET/SET) FLIP FLOP.

This is the simplest circuit capable of remembering the previous input:



To understand the operation of the flip-flop, remember that a NAND gate will give a TRUE output unless both its inputs are TRUE. Consider the following states:

State 1: $S=1$ and $R=0$, ie $\bar{S}=0$ and $\bar{R}=1$

If $\bar{S}$ is 0, Q is 1

Both inputs to gate 2 are 1, hence $\bar{Q}$ is 0.

If the inputs are then changed to $S=R=0$, the following state is entered:

State 2: $\bar{S}=\bar{R}=1$

The inputs to gate 2 are still 1, so $\bar{Q}$ is 0 still

$\bar{Q}$ is 0 so gate 1 produces 1.

If the inputs are then changed to $S=0$, $R=1$, then:

State 3: $\bar{S}=1$, $\bar{R}=0$

$\bar{R}=0 \Rightarrow \bar{Q}=1$

$\bar{Q}=1$ and $\bar{S}=1 \Rightarrow Q=0$.

If both inputs are again returned to 0, then:

<u>State 4:</u> $\bar{S} = \bar{R} = 1$

$\bar{S} = 1$ and $\bar{Q} = 1 \Rightarrow Q = 0$ still

$Q = 0 \Rightarrow \bar{Q} = 1$

<u>NOTE:</u> The outputs are always complementary, here Q and $\bar{Q}$

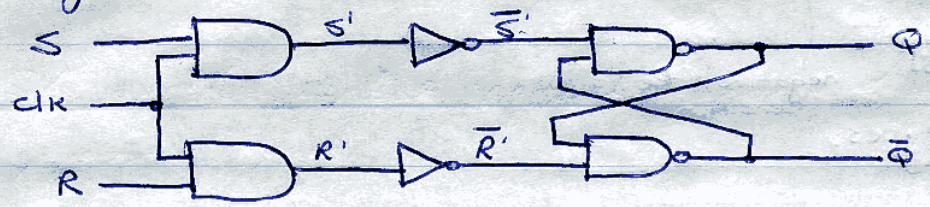
- If both R and S are 1 then both outputs will be 1. This is not a valid state as the outputs are not complementary. If R and S are then simultaneously changed to 0 then the values of the outputs would be indeterminate.

State	R	S	Q	$\bar{Q}$
1	0	1	1	0
2	0	0	1	0
3	1	0	0	1
4	0	0	0	1

We can summarise the RS flip-flop's operation as:
THE RS Flip-Flop remembers which of the inputs R or S was true last.

CLOCKED RS FLIP-FLOP.

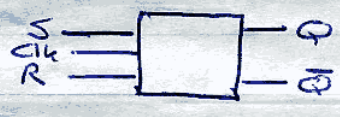
In a clocked RS FF the RS inputs can be blocked from changing the state of the FF by a control signal, the clock.



- If the clock input is 0 then both S' and R' will be zero and any changes in S and R will have no effect on the outputs.

This type of control line is called <u>level sensitive</u> as the logic level on the control line determines the state of the FF.

Schematic :

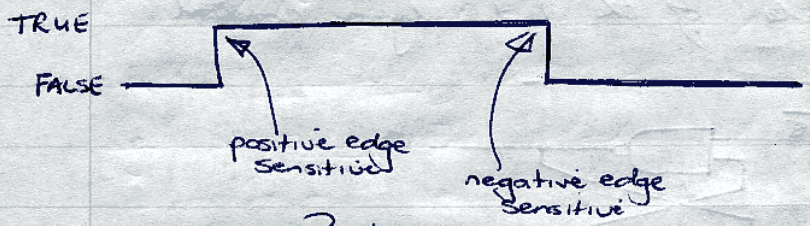


JK FLIP-FLOPS

To overcome the illegal mode of RS flip-flops, the JK flip-flop replaces the illegal input situation with a toggle mode, which inverts both outputs.

A level sensitive clock cannot be used with a JK flip-flop as if both inputs were high and the clock was high, the outputs would oscillate (and hence be effectively indeterminate). What is needed is a control line that is only active at a particular point on the clock cycle. The normal technique is to make the clock

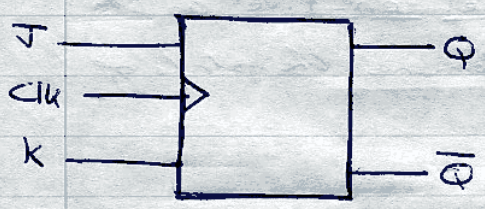
edge sensitive - only in transmissions from one logic state to the other in the clock will the inputs J and K be "active".



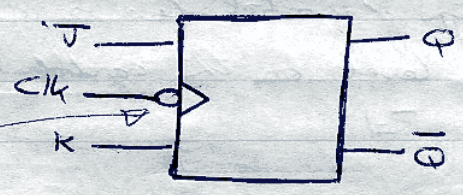
2 types:

Schematic

+ve edge sensitive



-ve edge sensitive



'>' ⇒ edge sensitive

'o' ⇒ control line is active low

D-TYPE FLIP-FLOPS.

If all we want to remember is the state of a particular logic variable then a D-type flip-flop would be used. All we need is a single input (the D or Data input), a clock line and an output or outputs.

We can construct a D type FF from a JK FF

Schematic

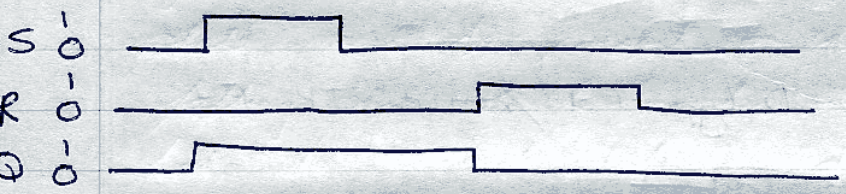
Operation of +ve edge sensitive D-type flip flop above:

"During a rising clock edge, the D input is transferred to the output and remains there till the next rising clock edge."

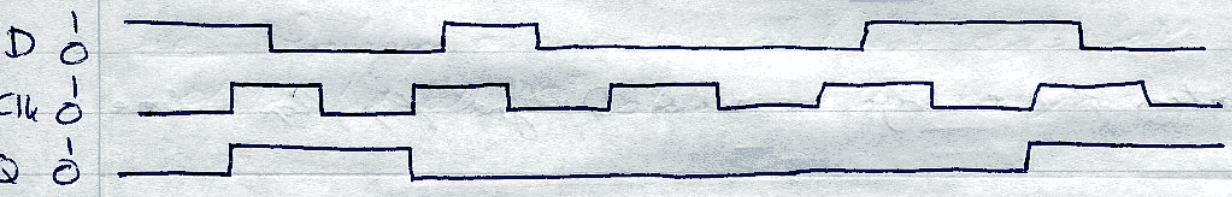
TIMING DIAGRAMS.

A timing diagram shows the states of the variables plotted against time.

Eg 1



Eg 2



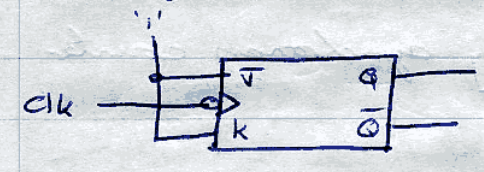
Note: changes in D have no effect on the outputs unless they coincide with a rising edge in the clock signal.

9 SEQUENTIAL DIGITAL CIRCUITS.

These are circuits in which the outputs are functions of the inputs and of the current state of the circuit.

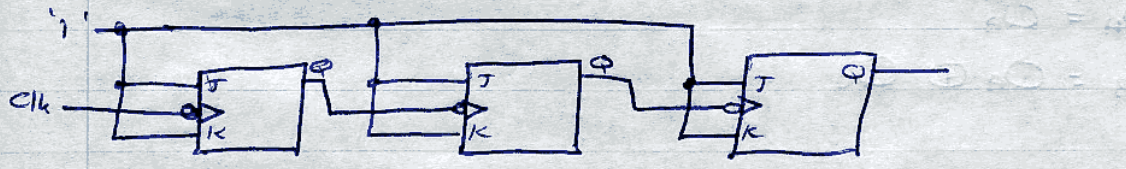
USING A J-K FLIP-FLOP IN TOGGLE MODE

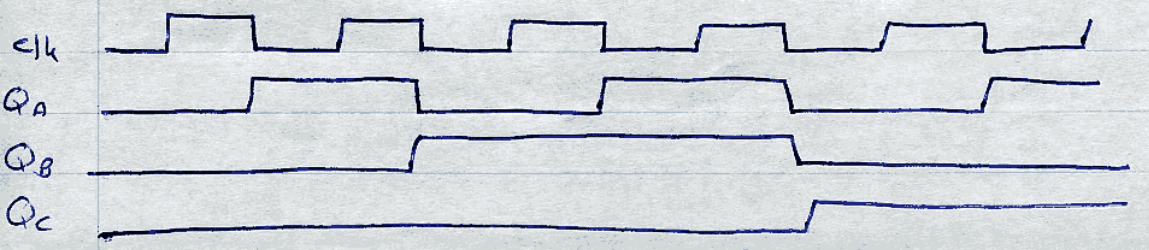
If both inputs of a -ve edge sensitive JK flip flop are connected to logical TRUE then, as every -ve edge on the clock will toggle outputs, the output will be a signal whose frequency is half that of the clock.



ASYNCHRONOUS COUNTERS.

If a number of JK flip flops with inputs tied to 1 are connected in series as below, they form an asynchronous natural binary counter, also called a <u>ripple counter</u> as a clock transition 'ripples' through from the left-most to right-most FF.





SYNCHRONOUS COUNTERS

Unlike a ripple counter where the clock input of each stage comes from the output of the previous stage, all the clock inputs of the SC are synchronised to the same master clock. If we want to design a synchronous counter using D-type FF's, then the next state must be present on the D inputs during the falling edge of the master clock. The basic idea is that the current state is fed into a code converter which gives as output the next state.

Eg: Design a 2-bit natural binary synchronous counter using D-type flip-flops.

Step 1: Draw up a present state / next state table. This should list all 2^2 states and the corresponding next state.

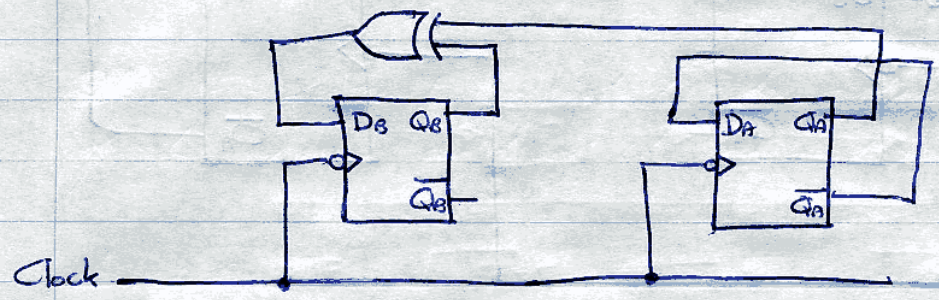
PRESENT STATE			NEXT STATE.		
DEC	Q_B	Q_A	DEC	D_B	D_A
0	0	0	1	0	1
1	0	1	2	1	0
2	1	0	3	1	1
3	1	1	0	0	0

Step 2: Simplify the functions.

$$D_A = \overline{Q_A}$$

$$D_B = Q_A \oplus Q_C$$

Step 3: Draw the circuit



SYNCHRONOUS COUNTERS WITH OTHER INPUTS.

The combinational logic circuit used to convert the present state into the next state can also have other inputs, used to control the count sequence.

Eg: Design a 2-bit synchronous counter that can count up or down, depending on whether the control variable U is TRUE or FALSE.

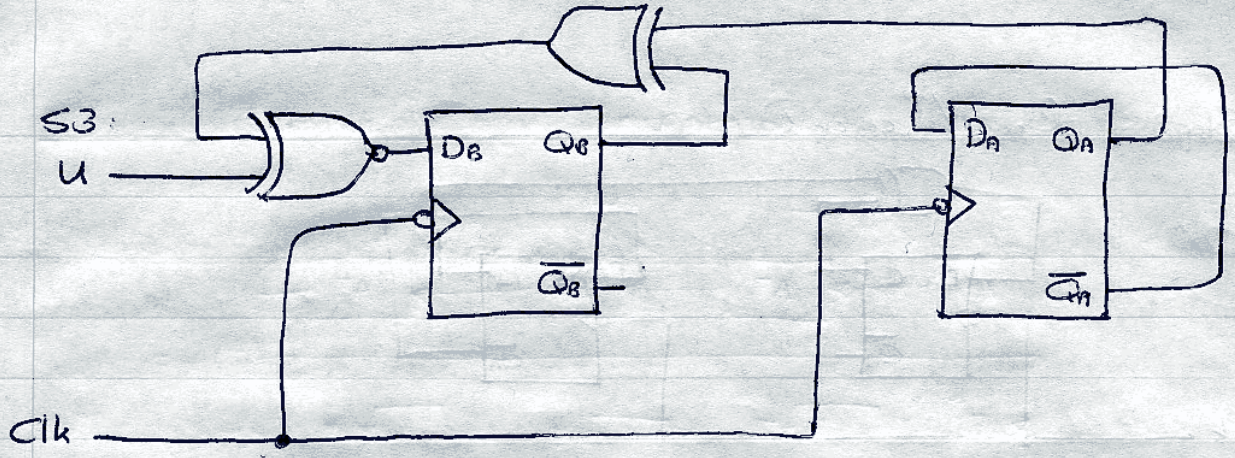
S1:

PRESENT STATE			COUNTER DIRECTION & PRESENT VALUE		NEXT STATE	
U	Q_B	Q_A			D_B	D_A
0	0	0	DOWN	0	1	1
0	0	1	DOWN	1	0	0
0	1	0	DOWN	2	0	1
0	1	1	DOWN	3	1	0
1	0	0	UP	0	0	1
1	0	1	UP	1	1	0
1	1	0	UP	2	1	1
1	1	1	UP	3	0	0

S2: Simplify

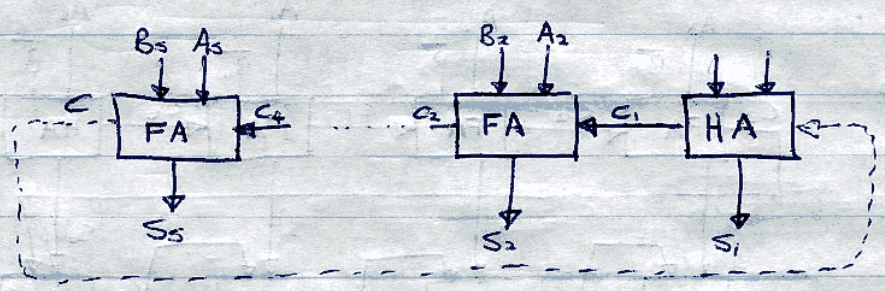
$$D_A = \overline{Q_A}$$

$$D_B = \overline{Q_A \oplus Q_B \oplus U}$$



(END OF 2ND YEAR DIG ELEC).

10 Implementation of Complement Arithmetic.

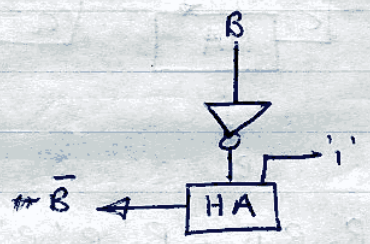


$A = A_5 A_4 A_3 A_2 A_1$ $\qquad$ $B = B_5 B_4 B_3 B_2 B_1$

In one's complement, we would have to add the carry bit. Either we could make the end adder a full adder with dotted line as shown (unstable, due to delay), or add further adders between C and S_1, etc.

2's complement is thus much more convenient.

To negate B:

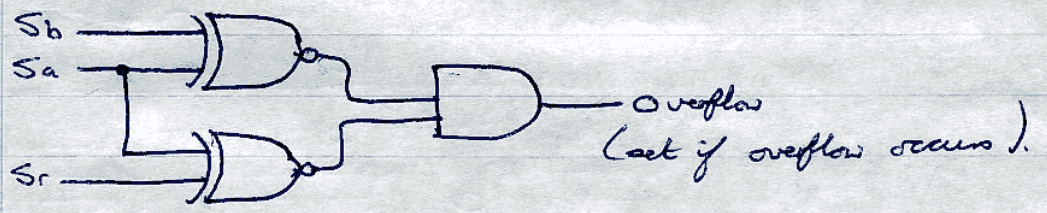


Overflow in 2's Complement.

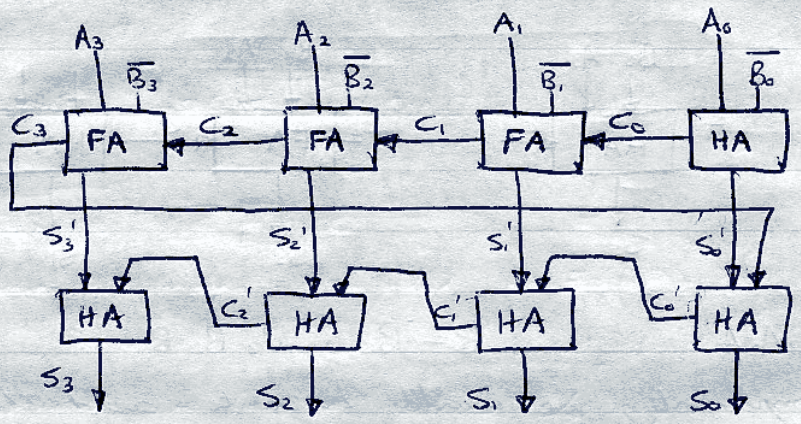
$$\begin{array}{r} S_b\ b_3\ b_2\ b_1\ b_0 \\ +\ \underline{S_a\ a_3\ a_2\ a_1\ a_0} \\ S_r\ r_3\ r_2\ r_1\ r_0 \end{array}$$

2 +ve numbers : overflow gives −ve result

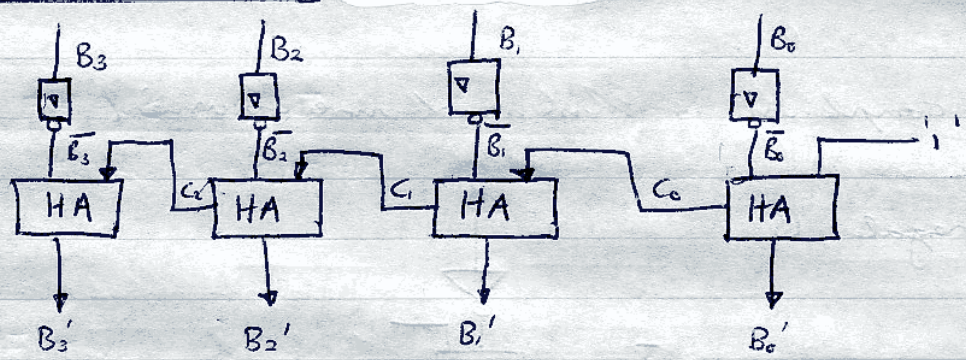
1 −ve number : overflow gives +ve result.



One's Complement Subtraction

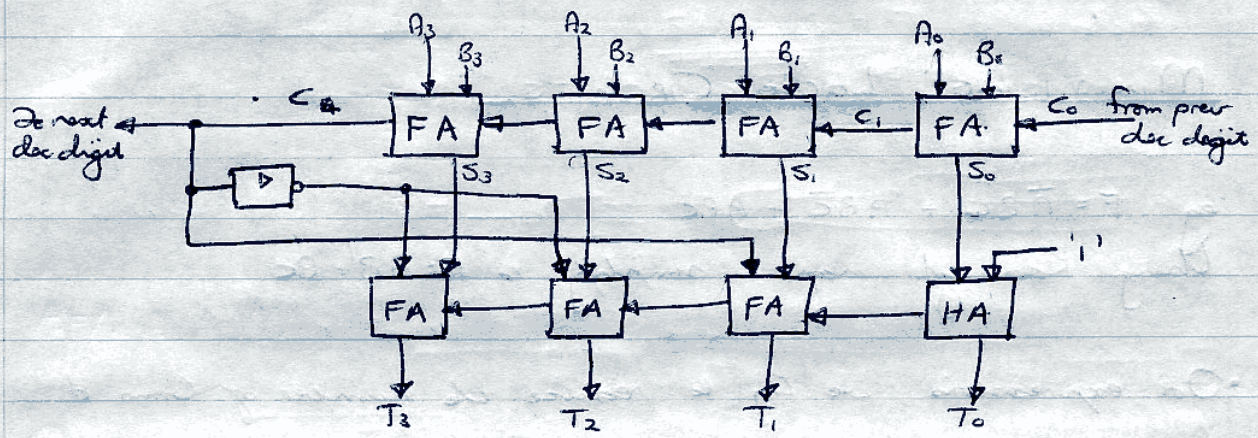


Two's Complementer



2's complement of $B_3 B_2 B_1 B_0 = B_3' B_2' B_1' B_0'$

XS-3 BCD ADDITION.



11. SEQUENTIAL & COMBINATIONAL LOGIC CIRCUITS.

MINIMISING LOGIC CIRCUITS.

eg $F = \bar{A}BC + AB\bar{C} + ABC$

Using Karnaugh maps, simplify to $F = AB + BC$

- For expressions, we try to reduce the number of terms & the number of factors in each term.

DEF
- For circuits, we try to minimise the total number of input gate leads.

Eg

$F = ABC + AB\bar{C}$ (8 inputs)

$= AB(C + \bar{C}) = AB$ (2 inputs)

$F = AB + AC$ (6 inputs)

$= A(B + C)$ (4 inputs)

In general, factorising simplifies design (by reducing the number of operations).

DESIGNING USING NAND GATES.

$\bar{A} = \overline{A \cdot 1} = \overline{A \cdot A}$

$A \cdot B = \overline{\overline{A \cdot B}}$

$A + B = \overline{\bar{A} \cdot \bar{B}}$

Any logic function can be broken down into a sum (disjunction) of fundamental products

Eg

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

C \ AB	00	01	11	10
0	1			1
1			1	

$F = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + ABC$ (From Truth Table)

$= \overline{\overline{\bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + ABC}}$

$= \overline{\overline{(\bar{A}\bar{B}\bar{C})}\,\overline{(A\bar{B}\bar{C})}\,\overline{(ABC)}}$ (By De Morgan)

$F = \underbrace{\bar{B}\bar{C} + ABC}_{7 \text{ inputs}}$ (By K.M.) $= \underbrace{\overline{\overline{(\bar{B}\bar{C})}\,\overline{(ABC)}}}_{7 \text{ inputs}}$ (a fn. using NAND)

So no loss of efficiency using NAND.

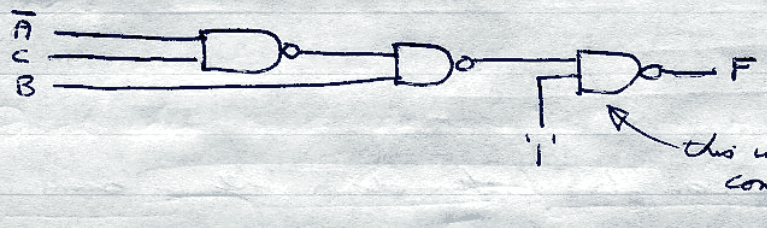
DEF SIMPLEST CIRCUIT USING ONLY NAND GATES is the one with the minimum number of inputs (assume all variables & complements are available : don't count inverters)

It is not always best to factorise, as conversion to NAND gates can increase complexity (see eg PTO)

Eg $F = A.B + B.\overline{C}$ (6 inputs)

factorise : $F = B(A+\overline{C})$ (4 inputs)

Convert : $F = B(\overline{\overline{A}.C})$



(6 inputs)

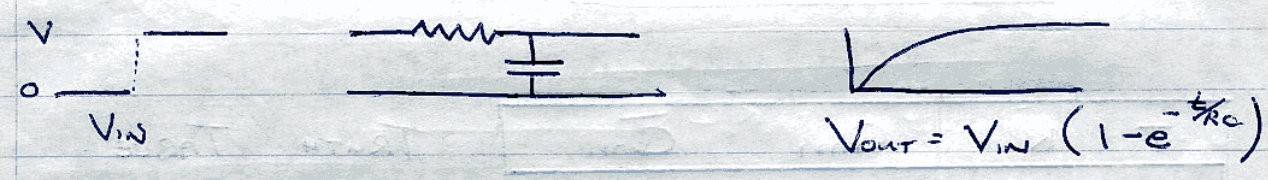
Converting the original equation: $F = \overline{\overline{A.B}\,.\,\overline{B.\overline{C}}}$ also 6 inputs.

IMPERFECTIONS IN DIGITAL CIRCUITS.

Ideally a digital system should switch between logic levels instantaneously (perfect square wave). In practice this is not the case.

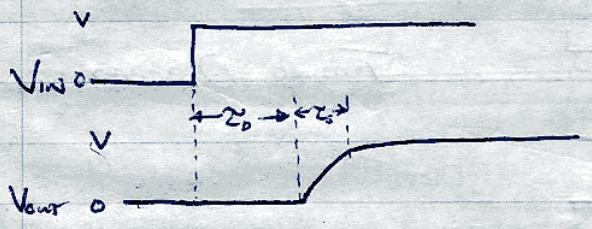
• SWITCHING TIME (τ_s)

Signals cannot change state instantaneously. Capacitive and inductive effect cause rounded edges.



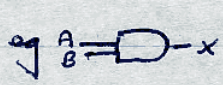
• PROPOGATION DELAY (τ_D)

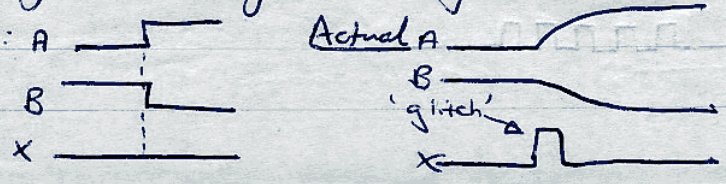
This is a pure delay between inputs changing and the outputs changing.



• RACING

If two inputs to a gate theoretically switch simultaneously we have a <u>race</u> condition. If switching was perfect and there were no propogation delays, racing would never occur.

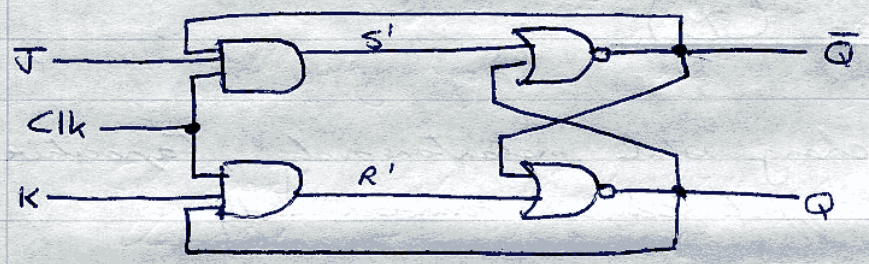
eg 



THE J-K FLIP FLOP

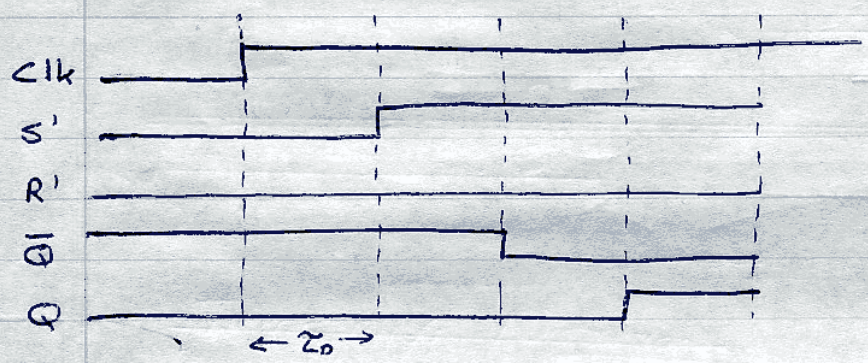
The invalid state (S=R=1) is replaced with a toggle mode.

First idea (not edge sensitive)



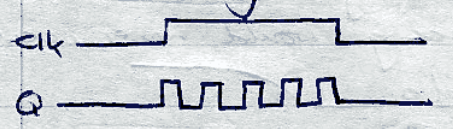
J	K	Q_{n+1}	$\overline{Q}_{n+1}$
0	0	Q_n	$\overline{Q}_n$
0	1	0	1
1	0	1	0
1	1	$\overline{Q}_n$	Q_n

TRUTH TABLE.



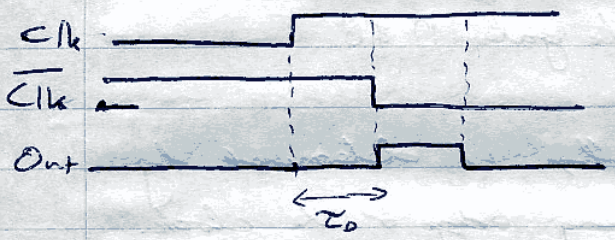
TIMING DIAGRAM.

The problem is that with the clock high, and J=K=1 the circuit will keep toggling, with the frequency depending on τ_D



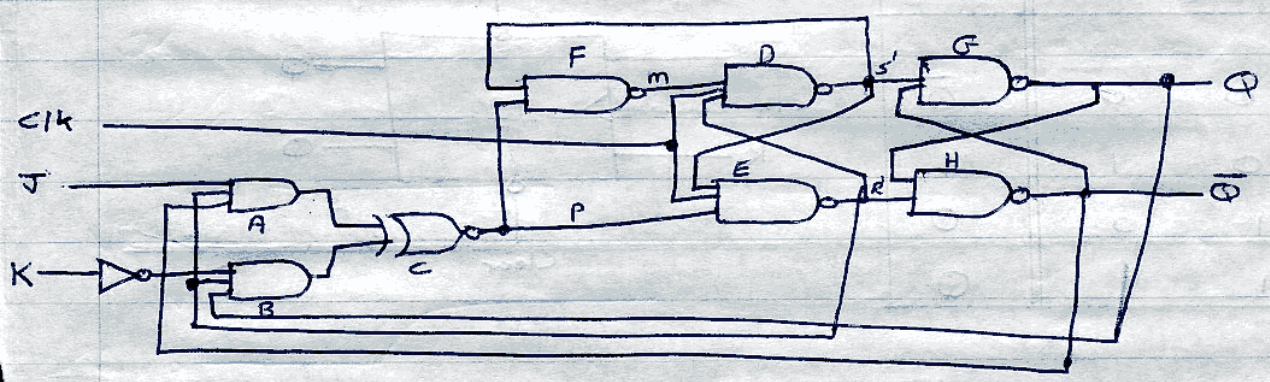
So we need an edge-sensitive FF.

First idea (BAD!)



This will not work - the clock must be high for two delays.

EDGE-TRIGGERED FLIP-FLOP.



+ve edge triggered J-K flip flop.

CASE 1: P = 0

Clk = 0 ⇒ S' = R' = 1, m = 1

The clock goes high....

all inputs to D are high ⇒ S' = 0, output flip flop GH set.

Even if P changes, the DE flip flop is blocked from changing.

Case 2: $P = 1$

$Clk = 0 \Rightarrow S' = R' = 1, \quad m = 0$

The clock goes high...
all inputs to gate E are high $\Rightarrow R' = 0$
(output F & GH reset)
P will stay high, R' goes into gate A & B.

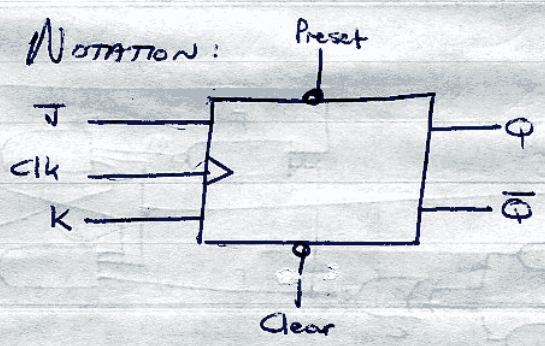
The next state of the flip flop is determined by the state of P at the rising edge of the clock.

$$P = \overline{J\bar{Q}R' + \bar{K}QR'}$$

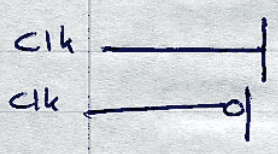
Note: if $Clk = 0$, $R' = 1$, so $P = \overline{J\bar{Q} + \bar{K}Q}$

J	K	P
0	0	$\bar{Q}$
0	1	1
1	0	0
1	1	Q

NOTATION:



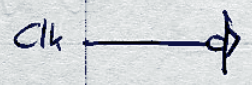
TRIGGERING NOTATION



level, active high

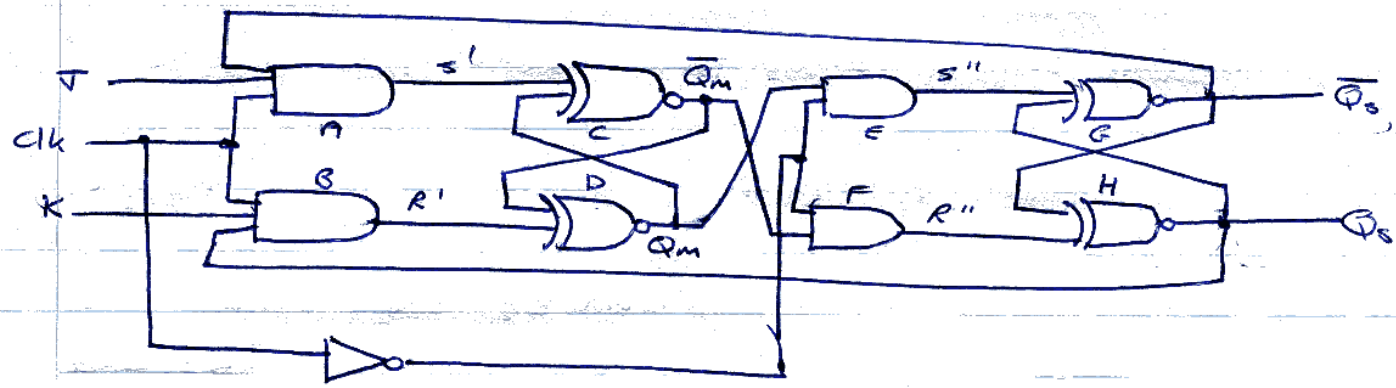
level, active low

edge, positive



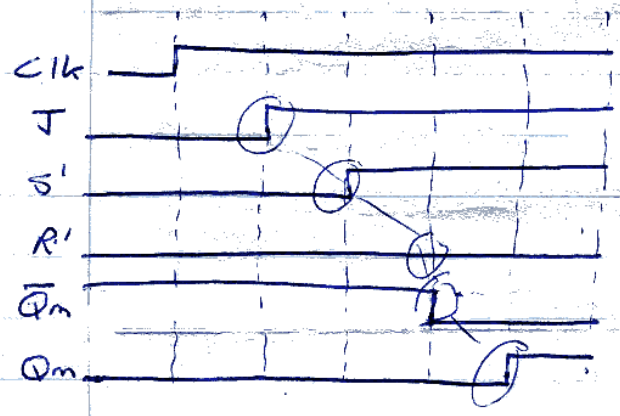
edge, negative.

MASTER-SLAVE FLIP FLOP



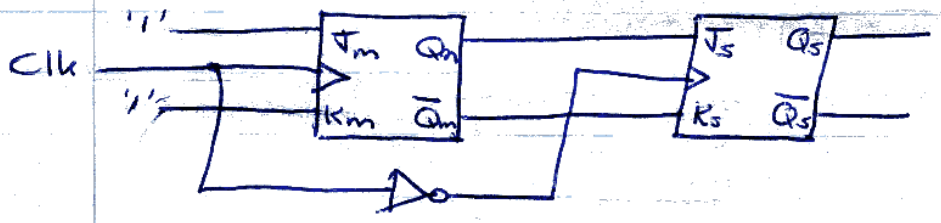
Problem: If J changes during the clock high then the flip-flop will be set (assuming it was RESET at the rising edge of the clock)

TIMING DIAGRAM:



So the output is not determined by the input at the rising edge

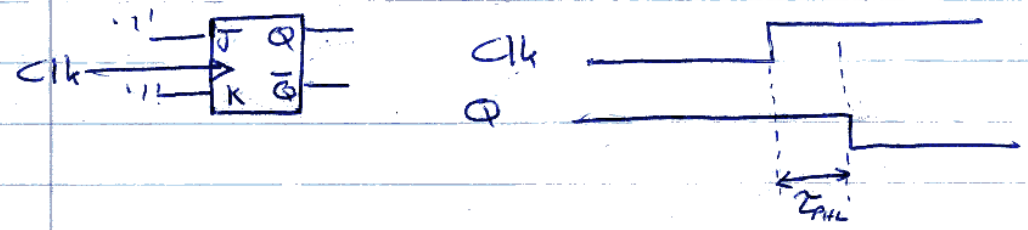
Solution:



SYNCHRONOUS COUNTERS

DEF Settling time of a counter: time taken for the counter to settle from the edge of the clock until the last flip-flop reaches its new value

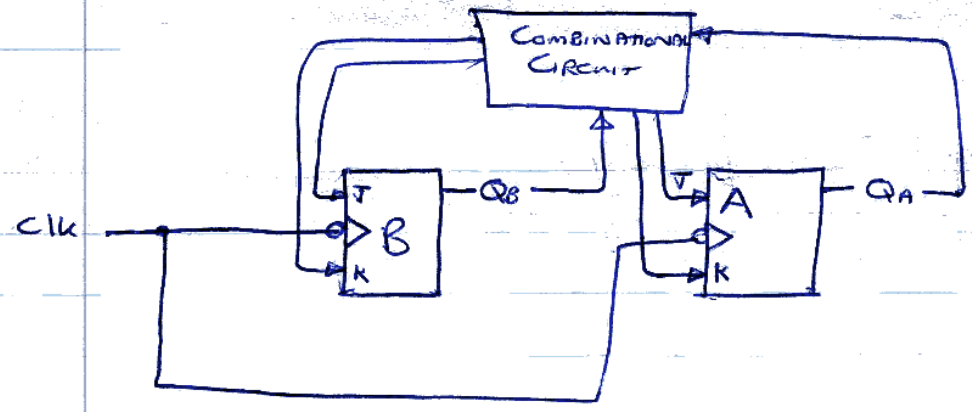
In an ASYNCHRONOUS COUNTER the settling time is $N\tau$, where N is the number of FF's and τ is the time taken for the clock to affect the output of a single FF.



The problem is caused by the fact that the effect of the clock ripples through the counter. The solution was a SYNCHRONOUS COUNTER, where all flip-flops have the same clock, with settling time τ.

SYNCHRONOUS COUNTER USING J-K FLIP-FLOPS

BASIC DIAGRAM



$J_A = f(Q_A, Q_B)$
$K_A = f(Q_A, Q_B)$
$J_B = f(Q_A, Q_B)$
$K_B = f(Q_A, Q_B)$

TRANSITION TABLE.

We have been using the following truth table to describe the function of the JK F.F.

J_n	K_n	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$

A transition table gives the values of J & K that are required to cause a particular transition in the output:

TRANSITION		VALUES	
Q_N	Q_{N+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

X = don't care condition

As before, we draw up a Present/Next state table, but we add the values of J and K to the table.

Example: MOD-5 COUNTER.

No	Present State Q_C	Q_B	Q_A	Next State No.	Q_C	Q_B	Q_A	J_C	K_C	J_B	K_B	J_A	K_A
0	0	0	0	1	0	0	1	0	x	0	x	1	x
1	0	0	1	2	0	1	0	0	x	1	x	x	1
2	0	1	0	3	0	1	1	0	x	x	0	1	x
3	0	1	1	4	1	0	0	1	x	x	1	x	1
4	1	0	0	0	0	0	0	x	1	0	x	0	x
5	1	0	1	x	x	x	x	x	x	x	x	x	x
6	1	1	0	x	x	x	x	x	x	x	x	x	x
7	1	1	1	x	x	x	x	x	x	x	x	x	x

Use KM to simplify J_C, K_C, etc.

$Q_B Q_A$ \ Q_C	0	1
00	0	x
01	0	x
11	1	x
10	0	x

$J_C = Q_A . Q_B$

$Q_B Q_A$ \ Q_C	0	1
00	x	1
01	x	x
11	x	x
10	x	x

$K_C = 1$

$Q_B Q_A$ \ Q_C	0	1
00	0	0
01	1	x
11	x	x
10	x	x

$J_B = Q_A$

$Q_B Q_A$ \ Q_C	0	1
00	x	x
01	x	x
11	1	x
10	0	x

$K_B = Q_A$

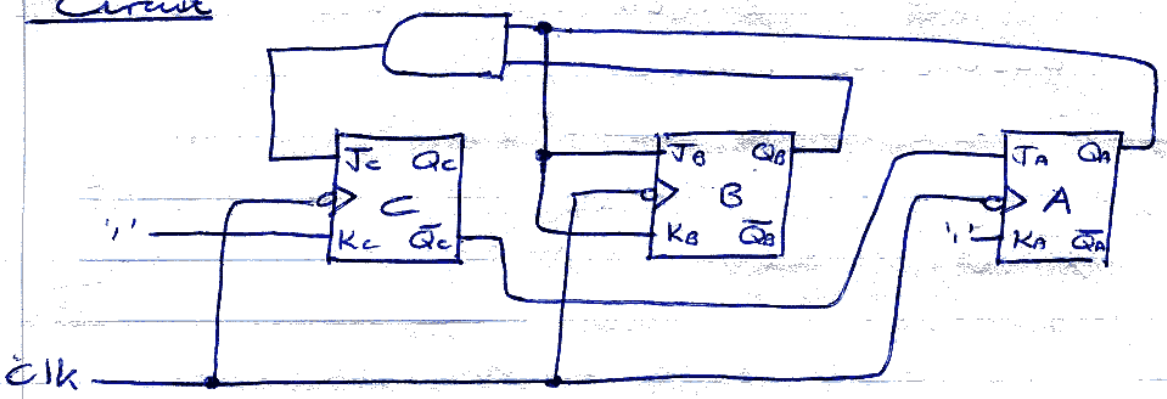
$Q_B Q_A$ \ Q_C	0	1
00	1	0
01	x	x
11	x	x
10	1	x

$J_A = \overline{Q_C}$

$Q_B Q_A$ \ Q_C	0	1
00	x	x
01	1	x
11	1	x
10	x	x

$K_A = 1.$

Circuit



DESIGNING USING A KEY MAP.

Example: Design a 2-bit Grey code counter.

PRESENT STATE			NEXT STATE						
No	QB	QA	No	QB	QA	JB	KB	JA	KA
0	0	0	1	0	1	0	X	1	X
1	0	1	3	1	1	1	X	X	0
2	1	0	0	0	0	X	1	0	X
3	1	1	2	1	0	X	0	X	1

- CREATE A KEY MAP:

QA \ QB	0	1
0	0	2
1	1	3

Label each block with its state number

- DRAW A KARNAUGH MAP FOR EACH J AND K.

QA \ QB	0	1
0	0	X
1	1	X

$J_B = Q_A$

QA \ QB	0	1
0	X	1
1	X	0

$K_B = \overline{Q_A}$

QA \ QB	0	1
0	1	0
1	X	X

$J_A = \overline{Q_B}$

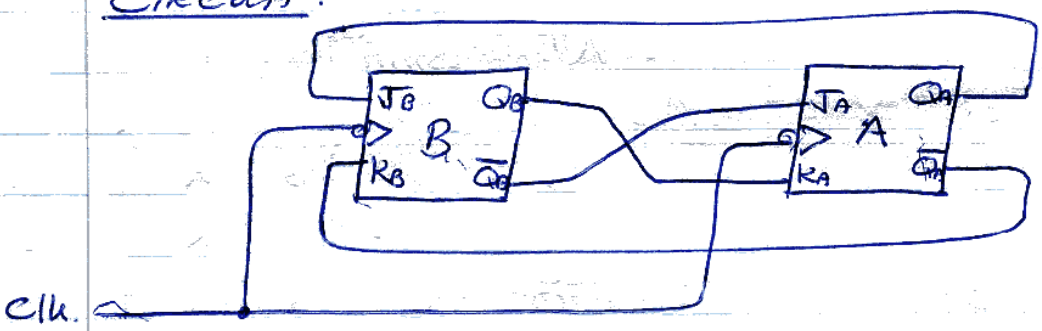
QA \ QB	0	1
0	X	X
1	0	1

$K_A = Q_B$

- FILL IN THE KARNAUGH MAPS USING THE FOLLOWING RULES

- for every 'present state', locate the relevant square on each Karnaugh map from the key map.
- use the state table to find out what transition the FF makes out of that state.
- use the transition table to find out the value of each J and K input necessary to cause that transition
- enter this value in the square.

CIRCUIT:

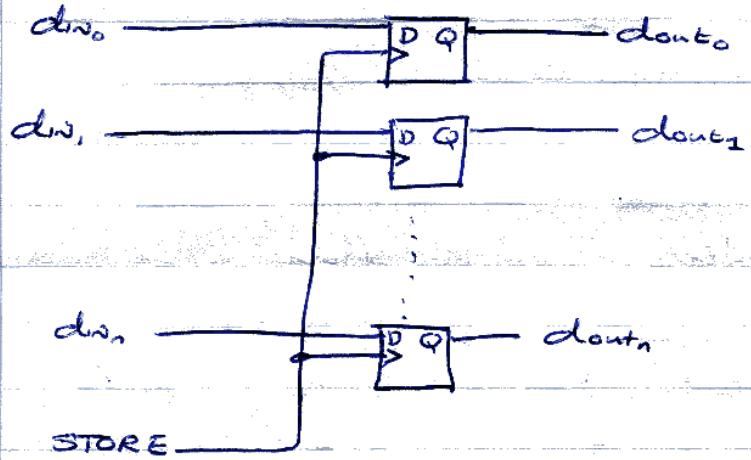


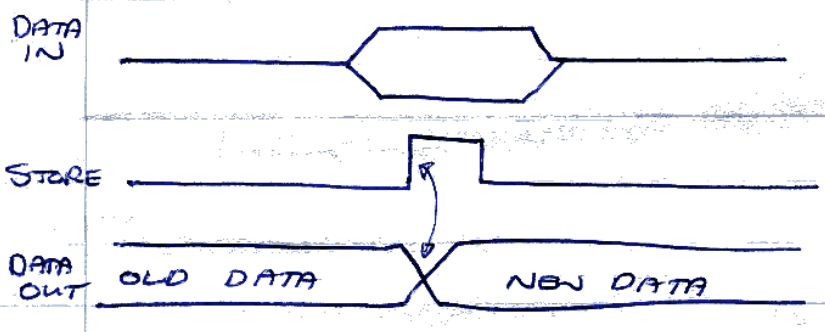
What to do with invalid states?

- next state is a dont care, or
- next state is a valid state (?)

REGISTERS.

A FF is a single bit memory.
A register is a number of flip-flops connected in parallel to form a multi-bit memory.

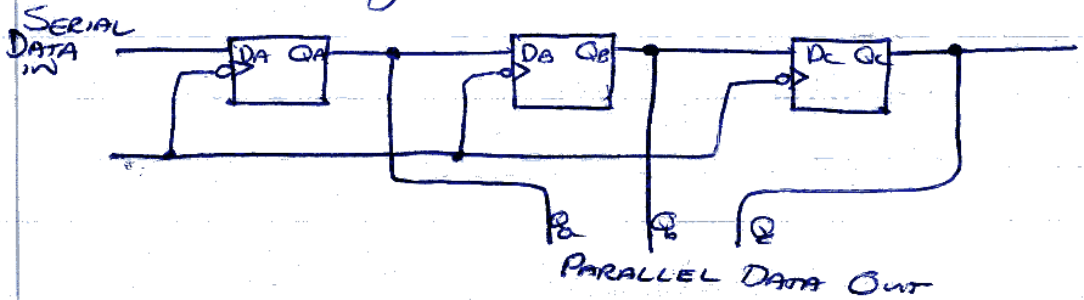




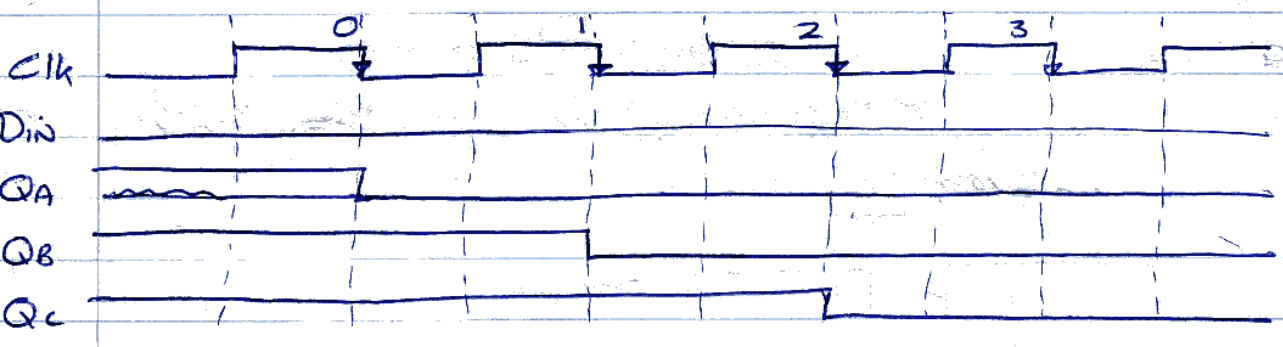
SHIFT REGISTERS.

SR's can be bit shifted left or right.
Uses: • arithmetic (multiplication & division by base)
• serial to parallel & vice-versa conversion.

Simple Shift register

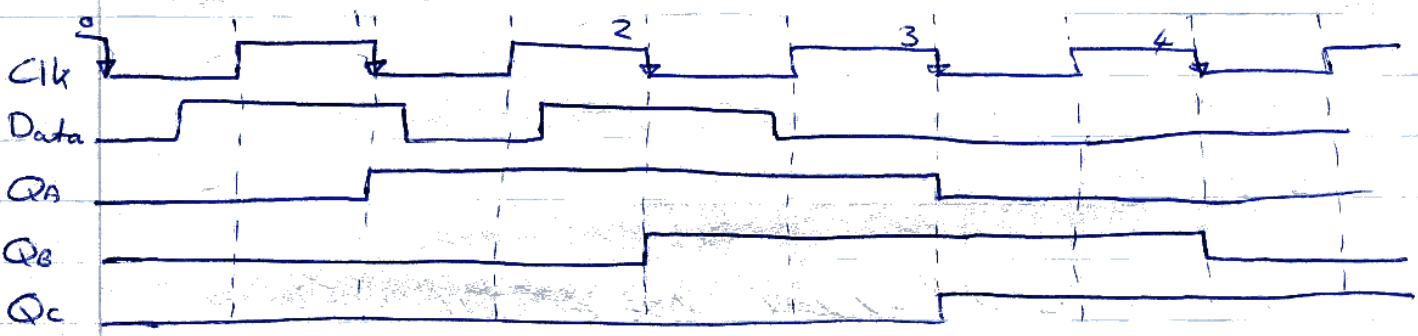


- Assume Register Preset to 110 ($Q_A Q_B Q_C$) no input data.



Clock Count	State Q_C	Q_B	Q_A	Binary Equivalent
0	0	1	1	3
1	1	1	0	6
2	1	0	0	4
3	0	0	0	0

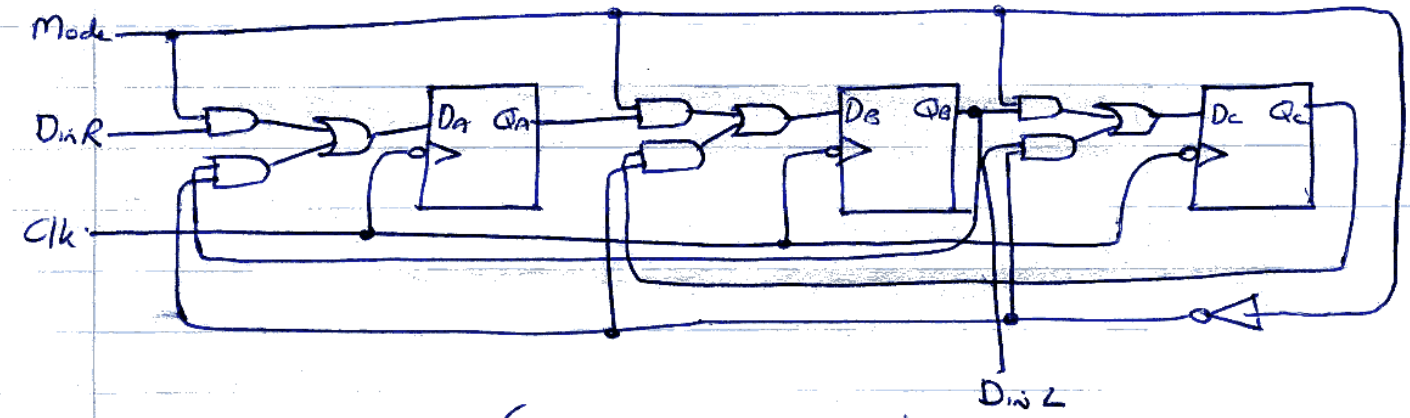
- Parallel to serial / serial to parallel conversion



Clock	State	Din at clock edge
0	000	0
1	000	1 } serial input bits.
2	001	1
3	011	0
4	110	0

Serial communications save on wire, are cheap, and are more reliable over long distances (no drift) but they are much slower than parallel.

BI DIRECTIONAL SHIFT REGISTER



If Mode = 1 (then shift right)

$m = 1 \Rightarrow$ $D_A = D_{in}R$
$D_B = Q_A$
$D_C = Q_B$

(all top and gates enabled and bottom ones disabled)

thus with each clock pulse the data will be shifted right along the register.

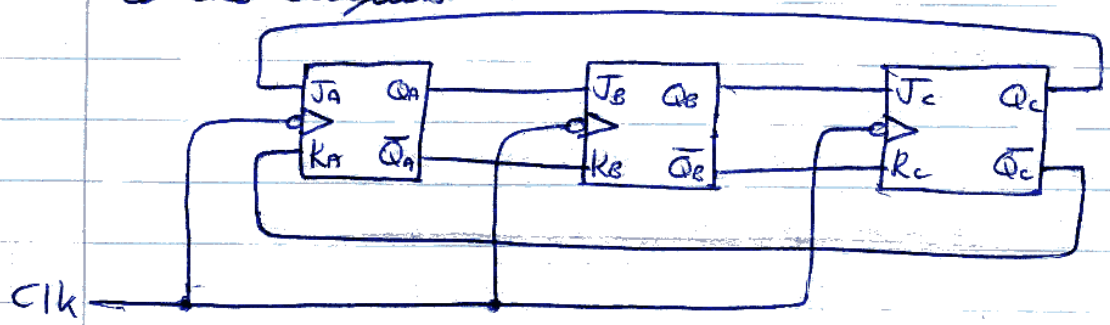
If Mode = 0 (Shift left)

(all top & gates disabled & bottom one enabled)

$m = 0 \Rightarrow$ $D_A = Q_B$
$D_B = Q_C$
$D_C = D_{inL}$

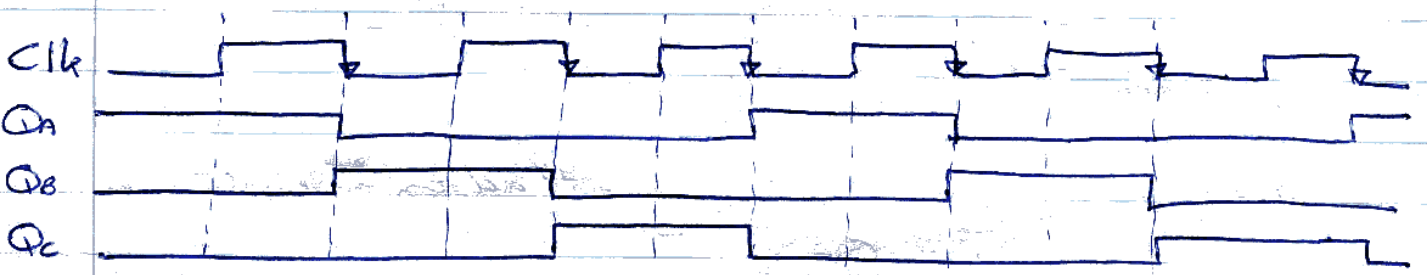
RING COUNTER.

A RC is a shift register with inputs connected to the outputs.



The counter must be preset to the desired pattern.

Eg $Q_A Q_B Q_C = 100$

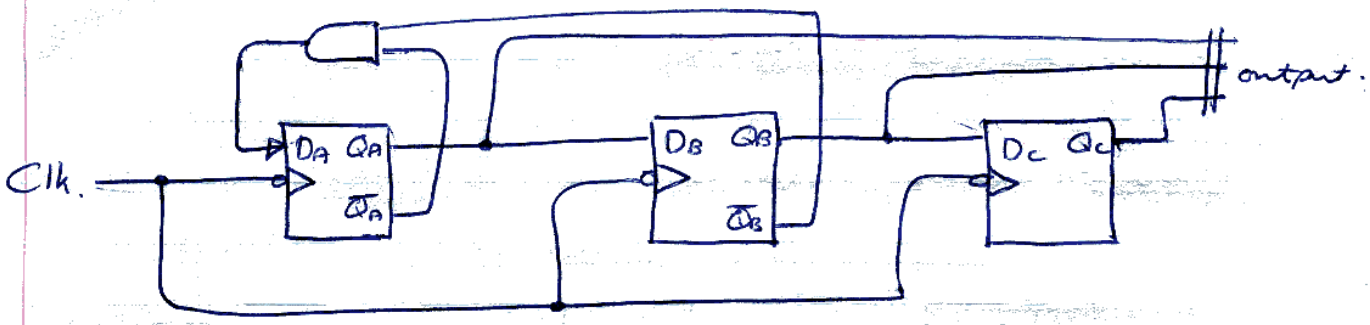


Q_A	Q_B	Q_C
1	0	0
0	1	0
0	0	1
1	0	0
0	1	0
0	0	1

etc.

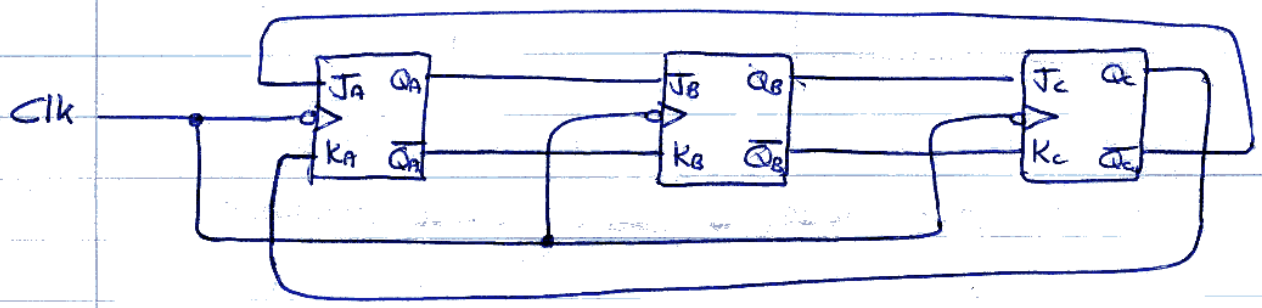
Problem: If the preset pattern is corrupted, then the corrupted pattern will continue to circulate in the counter.

Solution: a self-correcting ring counter.


output.
Clk.
D_A Q_A
$\overline{Q}_A$
D_B Q_B
$\overline{Q}_B$
D_C Q_C

JOHNSON COUNTER

Also called a "twisted-ring" or "switch-tail" counter.



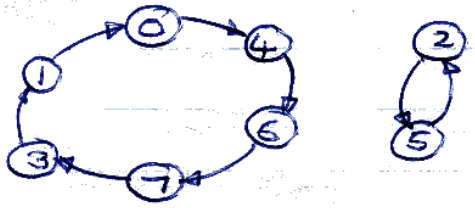
3-bit Johnson Counter.

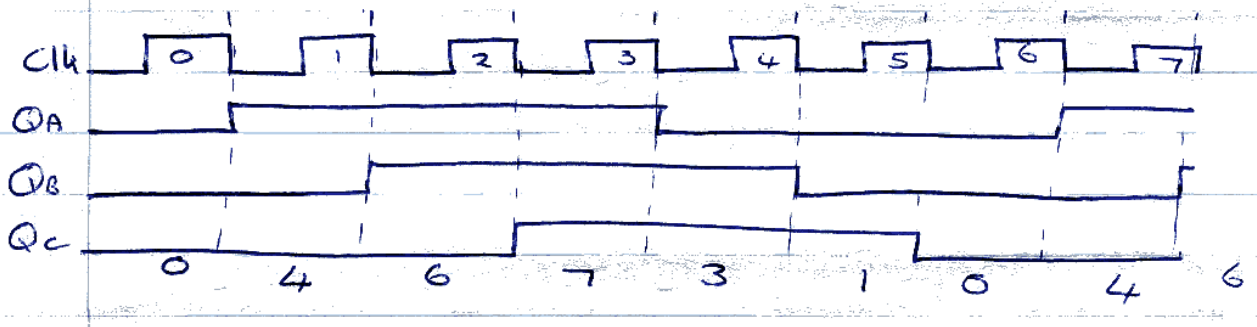
To derive the count sequence of the counter, use a "STATE CHECKOUT TABLE".
A S.C.T. works on the reverse principle to a synchronous counter design.

$J_A = \overline{Q_C}$ $\quad J_B = Q_A$ $\quad J_C = Q_B$

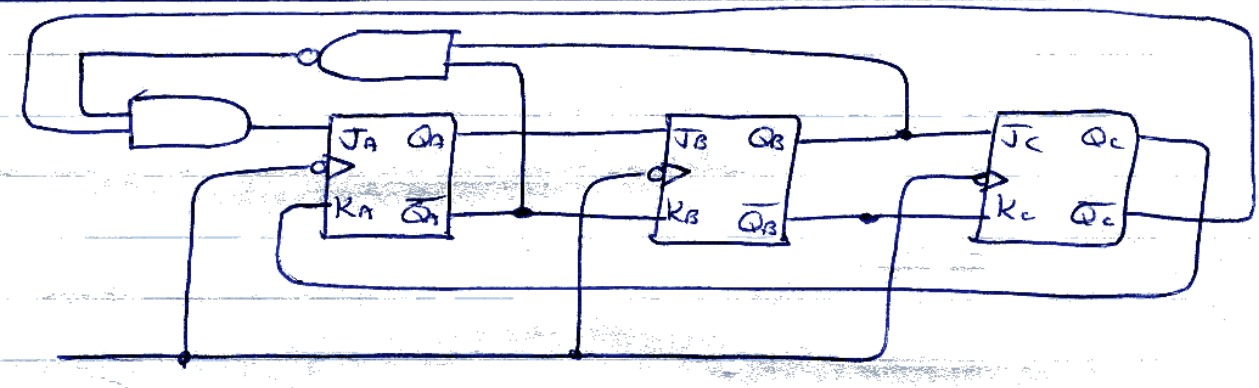
$K_A = Q_C$ $\quad K_B = \overline{Q_A}$ $\quad K_C = \overline{Q_B}$

	Present State			JK's			Next State			
No	Q_A	Q_B	Q_C	$J_A K_A$	$J_B K_B$	$J_C K_C$	Num	Q_A	Q_B	Q_C
0	0	0	0	10	01	01	4	1	0	0
1	0	0	1	01	01	01	0	0	0	0
2	0	1	0	10	01	10	5	1	0	1
3	0	1	1	01	01	10	1	0	0	1
4	1	0	0	10	10	01	6	1	1	0
5	1	0	1	01	10	01	2	0	1	0
6	1	1	0	10	10	10	7	1	1	1
7	1	1	1	01	10	10	3	0	1	1





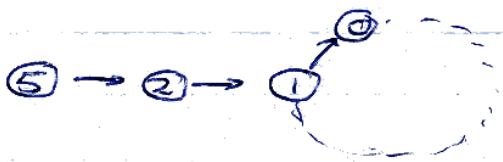
SELF-CORRECTING JOHNSON COUNTER.



$$J_A = \overline{Q_C} \cdot \overline{\overline{Q_A} \cdot Q_B} = \overline{Q_C}\,(Q_A + \overline{Q_B})$$
$$= \overline{Q_C}\,Q_A + \overline{Q_C}\,\overline{Q_B}$$

So J_A is 1 for $Q_A Q_B Q_C$ = 100, 110, 000, 100

000 → 100
001 → 000
(2) * 010 → 001 invalid state corrected.
011 → 001
100 → 110
(5) * 101 → 010
110 → 111
111 → 011



12 DIGITAL TECHNOLOGIES

12.1 We consider digital circuits implemented as integrated circuits, as IC's are more reliable and much cheaper to mass-fabricate than circuits using discrete components.

IC's which share a common design philosophy are grouped together in <u>logic families</u>, which determines many of the properties of an IC.

12.2 LOGIC CONVENTIONS

In binary digital circuits '0' and '1' are represented by two different electrical signal levels. These can be either current, or more typically, voltage. We distinguish between the signal levels by naming them HIGH and LOW. We thus have:

LOGIC STATE	POSITIVE LOGIC	NEGATIVE LOGIC
0	LOW	HIGH
1	HIGH	LOW

A change of convention affects the logic of our circuit. For example, if we have an AND gate with +ve logic, then <u>interpret</u> this as -ve logic, we are effectively inverting both inputs and outputs and thus have an OR gate. ⇒ convention is important.

12.3 CHARACTERISTICS & REQUIREMENTS OF A LOGIC FAMILY

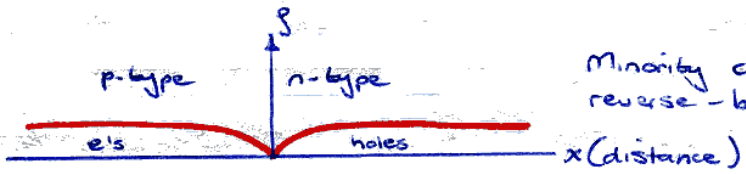
12.3.1 SPEED

Limited by two factors:

- RC time constants (cf pg 41)
- Switching speed of semi-conductor devices.

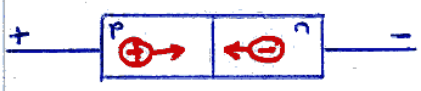
We first consider the switching speed of an ordinary pn diode. If the diode is reverse-biased, then since the majority carriers in the junction region are depleted, only minority carriers (electrons in p-type, holes in n-type) can carry current across the junction. Under stable conditions, however, the minority carrier density at the junction is zero, as the minority carriers cannot cross the junction due to the reverse-biased electric field.

Minority carrier conduction in reverse-biased diode

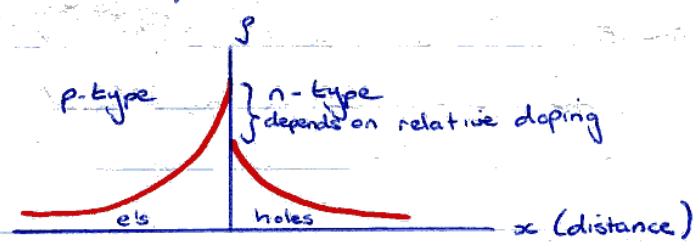


Minority carrier density in reverse-biased diode.

In a forward-biased diode, the current is carried across the junction by the majority carriers (holes in p-type, electrons in n-type). However, a steady-state minority carrier density also builds up:



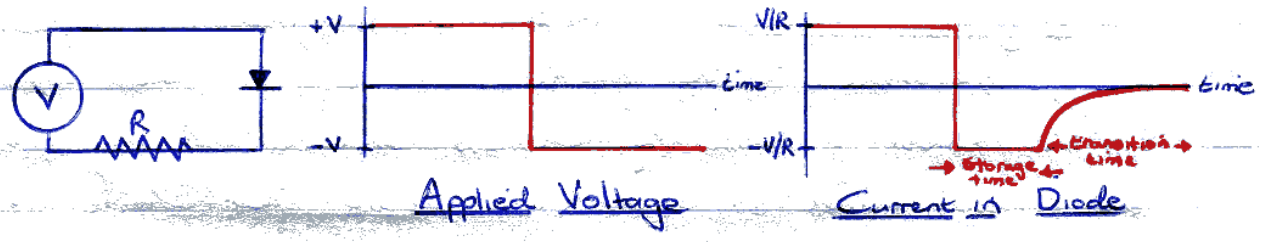
Majority carrier conduction in forward-biased diode



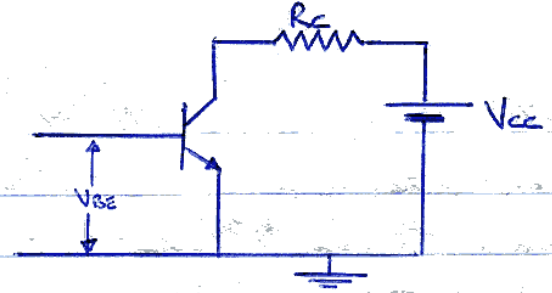
Minority carrier density in forward-biased diode

In the region of the junction the minority carrier density is high, as the majority carriers from the other side of the junction are swept across by the forward bias and become minority carriers once across the junction. These minority carriers will then attempt to combine with majority carriers, with greater success as they diffuse across the semiconductor, so the density drops off as we move away from the junction.

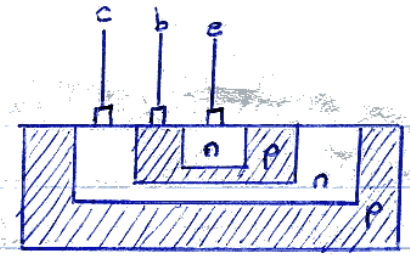
Suppose we instantaneously reverse the bias. Ideally we would want the diode current to drop instantaneously to zero; however, because of the minority carriers at the junction the current will instantaneously change direction and persist until all minority carriers have been carried across the junction. The time taken for this to happen is the storage time. When all the minority carriers are away from the junction, the current will not die away instantaneously as the junction capacitance must still be charged, which constitutes the transition time. The storage time is a pure delay, whereas transition time is an RC time constant.



The mechanism of switching in a junction transistor is also determined by minority carriers but in a somewhat different way. We shall consider an npn transistor connected as shown overleaf.



A npn-transistor switch circuit



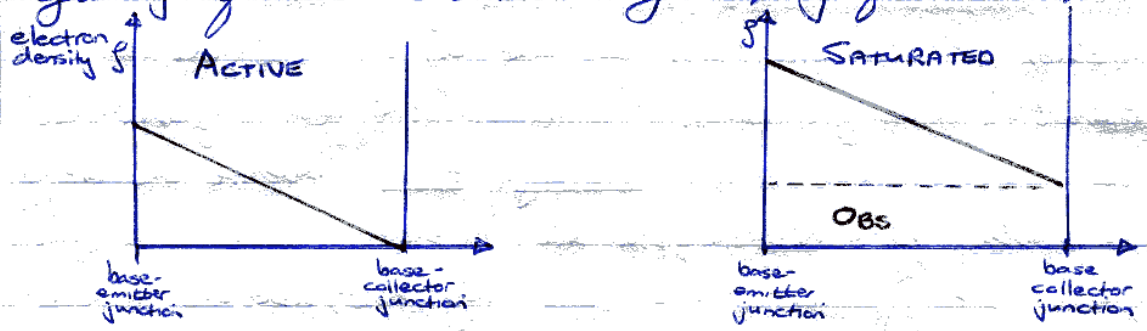
X-Section of an IC bipolar transistor

NB Emitter is small, heavily doped.
Base is a thin, lightly doped area

To turn the transistor on we forward-bias the base-emitter junction, causing electrons from the heavily-doped emitter to enter the base, where they are minority carriers. Very few holes leave the base to go to the emitter as the base is lightly doped. Thus to maintain charge neutrality in the base, positive charge must be supplied by means of a current from the base terminal. As the minority carrier density builds up at the base-emitter junction, a minority carrier gradient is built up across the base (as the base-collector junction is reverse-biased and thus has a minority-carrier density of ~0). Electrons will thus diffuse across the base region to the base-collector junction where they are swept across into the collector, causing the collector current, the magnitude of which is proportional to the gradient, which in turn depends on the number of minority carriers entering the base from the emitter, and hence on the positive base charge. The transition time to switch on the transistor is thus dependent on the size of the base current.

Having established a collector current, only a small base current is necessary to maintain it (to prevent the minority carrier density from declining due to recombination of electrons with holes in their passage across the base - this recombination is small due to the

geometry of the device and the light doping of the base.)



MINORITY - CARRIER DENSITY IN BASE OF A TRANSISTOR

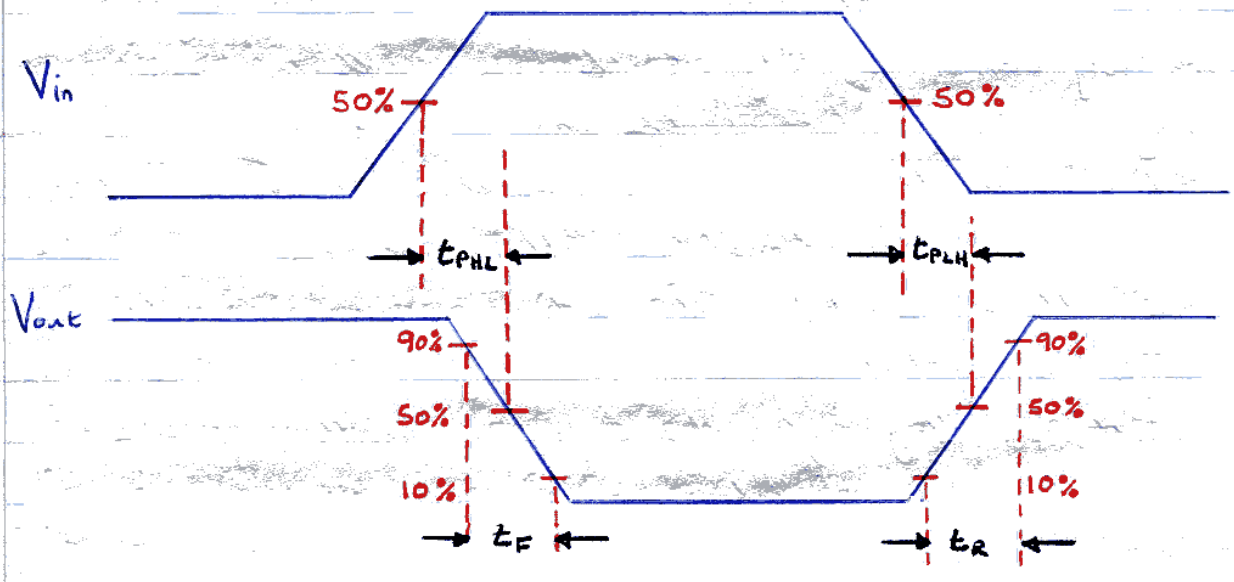
If we continue to increase the base voltage and current, the gradient, and hence the collector current will continue to increase (transistor in active region). However, as the collector current increases the collector potential decreases due to voltage drop across R_C; eventually the base-collector junction becomes forward biased, causing minority carriers to enter the base from the collector, raising the minority carrier density at the base-collector junction. Any further increase in the base current, and hence base charge, results in the m-c densities at both junctions to increase proportionately, with no increase in gradient (ie collector current). Thus when in saturation the transistor collects extra base charge Q_{BS}. The collector-emitter voltage of a transistor is small ($\sim 0.2V$) as is desirable for a switch.

To turn the transistor off, the collector current must be reduced to zero, so minority carriers must be removed from the base. If we simply removed the base potential the only way in which the positive base charge could be removed from the base would be through the recombination of holes and electrons. Since the base is lightly doped this is a relatively slow process so usually we reverse the base current.

If the transistor is in saturation there will be no reduction of the gradient until all the base saturation charge Q_{BS} is removed. This gives rise to a pure delay (the <u>storage time</u>) before the transistor can switch off. Thereafter the collector current begins to fall at a rate determined by how fast the reverse base current is removing charge from the base. Thus the <u>transition time</u> for switch off, as for switch on, is determined by the base current.

These delays are dependent on a complicated array of factors, so the manufacturer gives an overall set of specifications for the speed performance of the devices, given in terms of the values shown below:

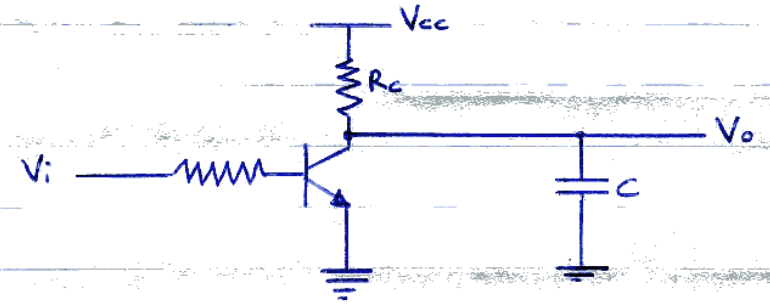
<u>SPEED SPECIFICATIONS OF A LOGIC GATE</u>



t_{PHL} = high-to-low transition propogation delay
t_{PLH} = low-to-high transition propogation delay
t_F = fall time
t_R = rise time.

12.3.2 Power Consumption

We generally try to minimise power consumption, especially for portable items. The power consumption of a family can also affect the upper limit on integration densities obtainable for the family. In general, we can trade-off power for speed, as illustrated below:



TRANSISTOR SWITCH DRIVING A CAPACITIVE LOAD

Here, charging time is approximately $R_C C$ when the transistor is switched off (a rough measure of switching speed). Thus a smaller value of R_C implies greater speed. The power consumption is approximately V_{CC}^2 / R_C, so a smaller value of R_C implies greater power consumption. The speed-power product $V_{CC}^2 \cdot C$ is independent of R_C, and shows that speed improvements (ie smaller value) result in greater power consumption, etc (ie switching time is inversely proportional to power consumption).

12.3.3 Circuit Complexity

The complexity of a circuit that can be achieved on a single silicon chip is dependent on 3 factors:

- production yield (a larger chip size has a greater chance of containing imperfections)
- geometric area of components (american: "Real estate") (passive devices occupy more chip area than active devices)
- power consumption (device heat output is limited by maximum heat dissipation ⇒ resistors undesirable)

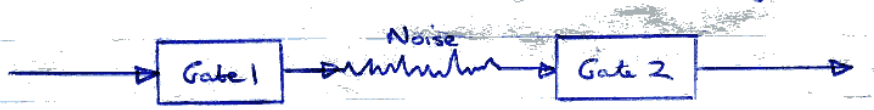
Circuit complexities are conventionally classified as:

- SSI Small-scale integration < 100 components/chip
- MSI Medium-scale integration 100-1000 " "
- LSI Large-scale integration 1000-10000 " "
- VLSI Very-large-scale integration > 10 000 " "

Note that a typical logic circuit consists of about 10 components.

12.3.4 NOISE IMMUNITY

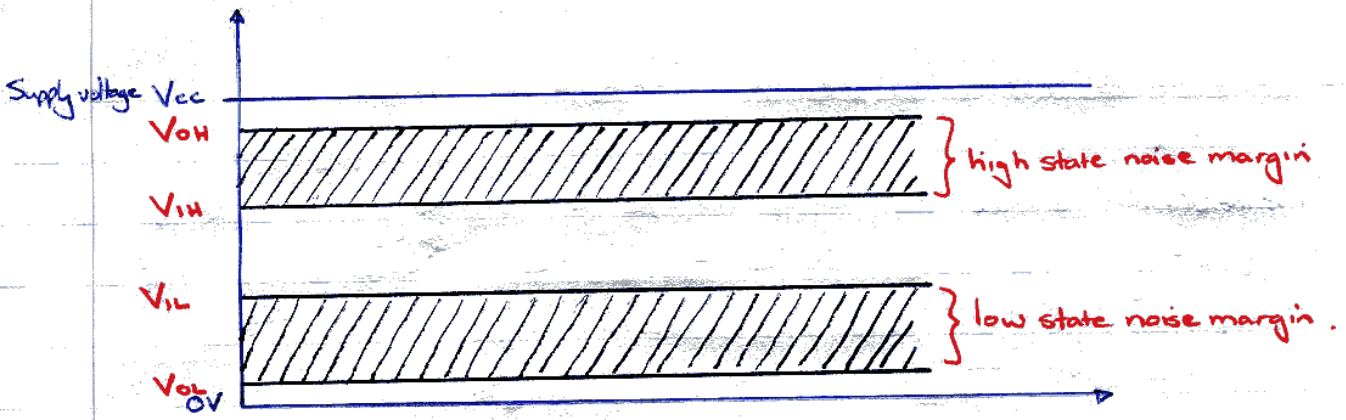
All circuits are subject to electrical noise from the environment which can lead to invalid circuit operation. Consider the output of one gate driving the input of another gate in the same logic family:



The manufacturers guarantee four levels:

V_{OL} - maximum output voltage in low state
V_{IL} - maximum acceptable input voltage recognised as low state
V_{IH} - minimum acceptable input voltage recognised as high state.
V_{OH} - minimum output voltage in high state.

We thus have a high state noise immunity margin of $V_{OH} - V_{IH}$, and a low state " " " of $V_{IL} - V_{OL}$. (see opposite)



2.3.5 FAN - OUT

The maximum number of inputs that can be driven by a single output (ie gate) is the fan-out of the logic family. The fan-out of a logic family is obtained in terms of the following manufacturer supplied figures:

I_{OH} = maximum current that the output will sink in the high state without the output voltage falling below V_{OH}

I_{OL} = maximum current that the output will sink in the low state without the output voltage rising above V_{OL}

I_{IH} = maximum current that the input will draw when the input voltage is V_{OH}

I_{IL} = maximum current that the input will draw when the input voltage is V_{OL}

NB:
- current into a terminal is defined as +ve irrespective of whether it is an input or output terminal.
- the currents above refer to steady-state currents: it is much more difficult to describe fan-out when considering switching currents.

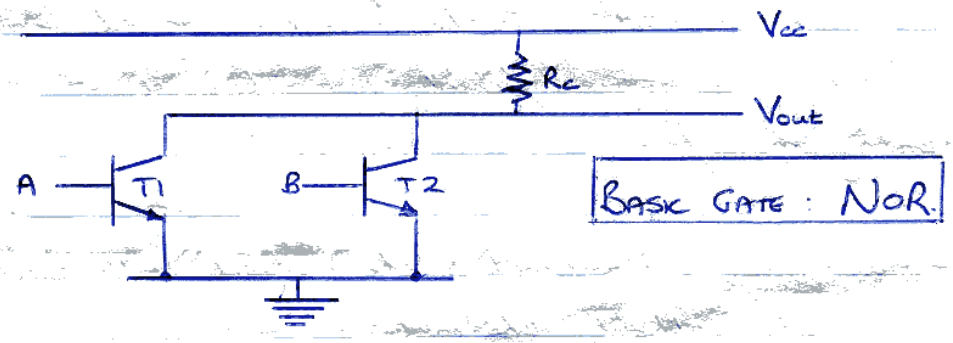
$$\text{FAN-OUT} = \min\left\{\left[\frac{-I_{OH}}{I_{IH}}\right], \left[\frac{-I_{OL}}{I_{IL}}\right]\right\}$$

2.3.6 Temperature Range

Normal industrial : 0°C – 70°C
Military – 55°C – 125°C.

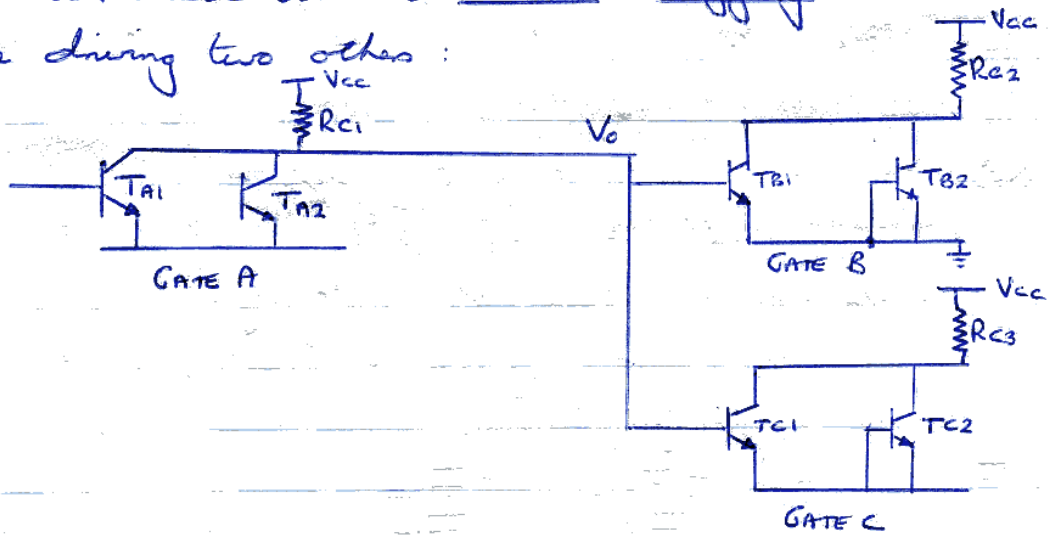
12.4 Resistor - Transistor Logic (RTL)

2.4.1 Direct - Coupled Transistor Logic (DCTL)



If A, B are low : T1, T2 off, so Vo pulled high by Rc
If either/both A, B goes high : then T1 and/or T2 switches on into saturation (if Rc is small enough) so Vo is low.

DCTL is never used due to current - hogging. Consider a NOR gate driving two others :

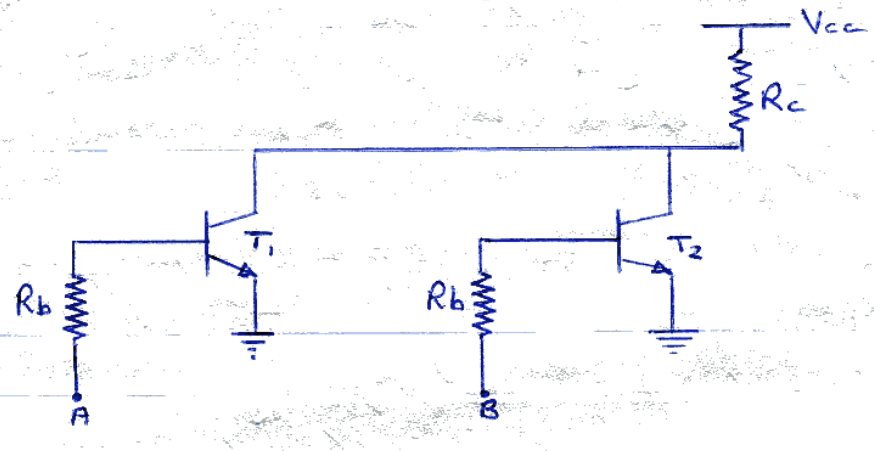


The transistors in gates B & C whose inputs are grounded are off and have no effect on the circuit: they can be ignored. Thus gates B & C behave as inverters driven by V_o, the output of gate A.

We assume that T_{B1} & T_{A2} are off, thus V_o is high. For correct operation, transistors T_{B1} and T_{C1} should then be driven into saturation, each requiring a minimum base drive current provided through resistor R_{C1} of gate A. If both transistors saturate at an identical V_{BE}, providing resistor R_{C1} is sufficiently small, then there is no problem. However, it is much more likely that transistors T_{B1} and T_{C1} saturate at different V_{BE}'s.

Assume that T_{B1} saturates at $V_{BE} = 0.74\,V$ and T_{C1} at $V_{BE} = 0.76\,V$, and assume that initially gate A is only driving T_{B1}. If R_{C1} is small enough to drive T_{B1} into saturation then the voltage at V_o is set by the saturation base-emitter voltage of T_{B1} (ie 0.74 V). If we now connect up gate C, transistor T_{C1} will be in its active region. In order for T_{C1} to saturate, V_o would have to rise to 0.76 V and thus the current through R_{C1} would have to drop. However, this current is saturating T_{B1} and thus cannot fall off as this would require the base voltage of T_{B1} to fall, a contradictory requirement. So T_{B1} is "hogging" all the current delivered by R_{C1} and thus there is no base drive current for T_{C1}; the output of T_{C1} is not low and the operation is invalid.

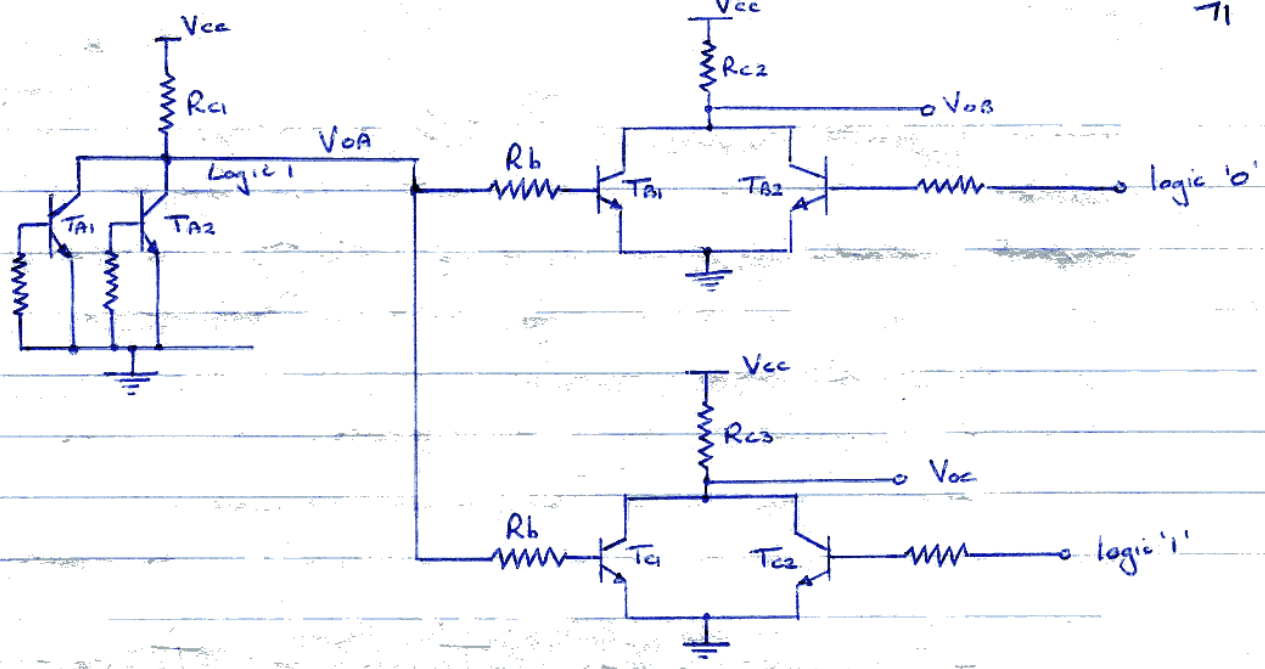
12.4.2 RESISTOR - TRANSISTOR LOGIC (RTL)



BASIC RTL NOR GATE.

If we insert a resistance into the base circuit of each transistor, we can eliminate DCTL current hogging - this is called RTL. Current hogging is prevented by limiting the base current drawn by any transistor. Referring to our previous discussion, assume as before that initially T_{C1} is disconnected. Then, providing $R_{C1} + R_b$ is small enough, T_{B1} will be saturated. But if R_b is large enough w.r.t. R_{C1}, then V_o will be $> 0.76\,V$, so that when T_{C1} is connected some of the current provided by A through R_{C1} must be drawn into the base of T_{C1}. So providing gate A can supply enough current (ie R_{C1} is small enough) both T_{B1} and T_{C1} will be saturated despite their differences in $V_{BE(SAT)}$.

The main disadvantages of RTL are its low noise margin and low fanout: and hence its rare use.



EXAMPLE OF FAN-OUT PROBLEMS IN RTL

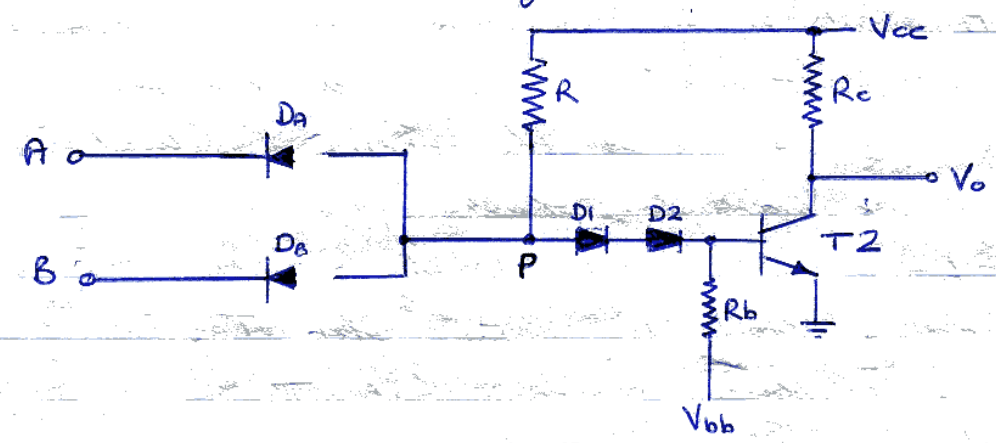
FAN-OUT: The output current in high state is provided through R_{C1}. The input current drawn by a gate with H input is the base current required to saturate a transistor - relatively large and thus limiting fanout. There is also the peculiar situation that if one transistor of the gate is already in saturation, the current drawn by the other becomes much larger. Say V_{OA} is H. T_{B2} is off and thus has no effect on T_{B1}. But T_{C2} is in saturation so V_{OC} is about 0.2 V. When the base voltage of T_{C1} reaches turn-on voltage, T_{C1} immediately goes into saturation since the BC-junction will already be forward biased (due to low V_{OC}) so T_{C1} has no active region. If T_{B1} and T_{C1} were identical transistors, the base voltage of T_{C1} would be much lower than that of T_{B1} in this case. Since V_{OA} is common and the current drawn is determined by voltage drop across the Rb's, T_{C1} draws a higher current than T_{B1} (in practice up to 3 times as much). Guaranteed fanout only 5 gates

LOW NOISE IMMUNITY The output voltage in the low state is $V_{CE(sat)}$, about 0.2 V. To switch the input transistor of a gate on we need about 0.7 V, so a noise spike of only 0.5 V will cause invalid operation. In practice the noise margin is even less due to current loading, temperature, etc.

12.5 DIODE-TRANSISTOR AND TRANSISTOR-TRANSISTOR LOGIC

12.5.1 DIODE-TRANSISTOR LOGIC (DTL)

Diodes are used instead of resistors at the inputs.



BASIC DTL NAND GATE

The gate can have more inputs added by simply connecting further diodes such as DA and DB.

Operation: If either or both inputs are low (say A low) then DA is forward biased and current flows from Vcc through DA into the output of the driving stage. Assume the output voltage of the driving gate to be $V_{CE(SAT)}$ is about 0.2 V. Then the voltage at P is about 0.2 V + 0.75 V (the forward-

biased diode voltage drop). Thus T_2 must be off since to switch it on would require both diodes D1 and D2 to be forward biased (0.75V + 0.75 V) and V_{BE} of T_2 to be about 0.7V (transistor turn on voltage) ie voltage at P of at least ~2.2V. Since T_2 is off the gate has high output.

If both inputs are high (or unconnected) current cannot flow through D_A and D_B so it must flow through D1 and D2, switching on T_2 into saturation, so the output is $V_{CE(SAT)}$, about 0.2 V (ie low).

To turn T_2 off, we need Rb and Vbb. If Rb is too small, turn-on is slowed down. If one input goes low, the voltage at P drops to about 0.95 V and D1 and D2 are cut off. To switch off T_2 we need to remove the charge from the base. Vbb can be a negative voltage to facilitate this removal.

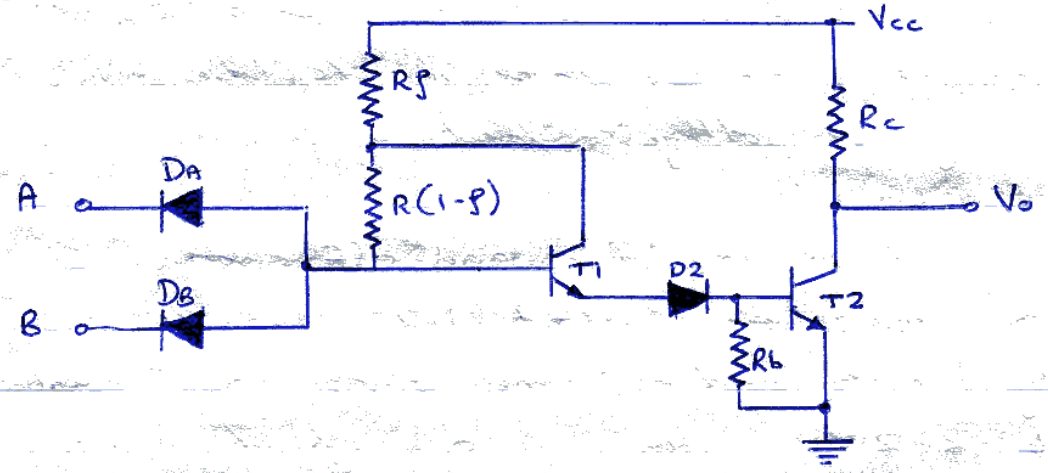
D1 and D2 are called stand-off diodes. They give a <u>noise immunity</u> of about 1.25 V. This is because if they are to conduct, the voltage at P must be about 2.2 V; when either or both the inputs is low the voltage at P is about 0.95 V.

The fanout capability of DTL is determined by its ability to sink current when the output is low, since for a H output the current drawn at the inputs is the reverse-bias current of a diode which is negligible. For L output, assuming only one input of each gate is driven low, then the input current of each driven gate is the current drawn by R, approx $\frac{V_{CC} - \overset{0.75}{V_D} - \overset{0.2}{V_{CE(SAT)}}}{R}$ where V_D is forward-biased input diode voltage drop and $V_{CE(SAT)}$ the output voltage of driving gate

We could thus apparently improve fanout by increasing R. However, the fanout is also determined by the output current sinking capability of the driving gate, determined by collector current of T2, in turn determined by base current. The base current is approx $\frac{V_{CC} - \overset{\approx 0.75}{V_{BE}} - 2\overset{\approx 0.75}{V_D}}{R}$

if we ignore I_{R_b}. By increasing R we reduce the base drive to T2 and hence reduce its ability to sink current. R is thus a tradeoff value.

In an IC implementation we use a transistor T1 in its active region instead of D1. This gives the same base-drive current for larger values of R, reducing input current and hence improving fanout. (typically fanout is 8 for IC DTL).

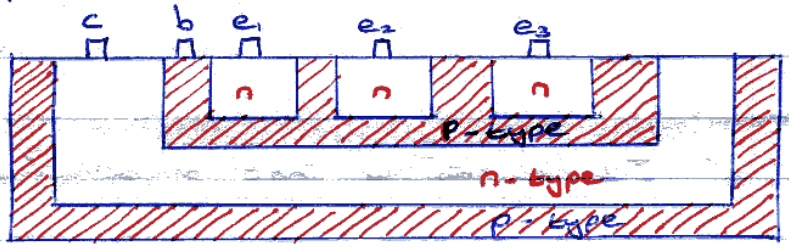


INTEGRATED CIRCUIT DTL NAND GATE

The main drawback of DTL is the slow switch off time. Decreasing Rb speeds up switching but decreases fanout as the base drive current of T2 drops. This problem is effectively overcome with TTL logic.

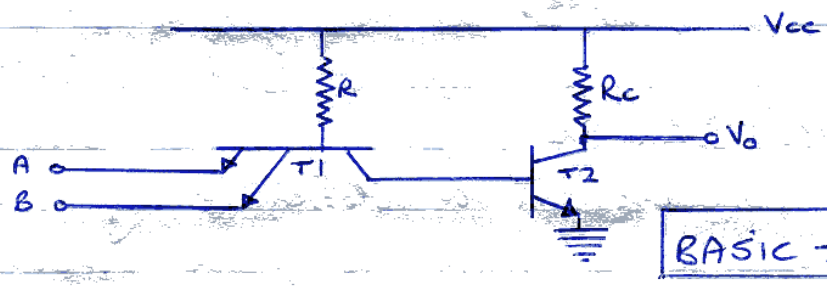
12.5.2 TRANSISTOR - TRANSISTOR LOGIC (TTL).

The switching speed of DTL is improved in TTL by replacing the input diode network with a multi-emitter transistor, formed by integrating a number of emitters in the same base.



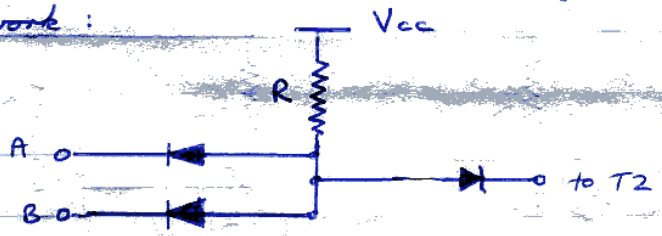
CROSS-SECTION OF IC MULTI-EMITTER TRANSISTOR

Main advantage: functional equivalence for much lower real-estate



BASIC TTL NAND GATE

The logical operation of TTL is identical to DTL, as the multi-emitter transistor is <u>statically</u> equivalent to this diode network:

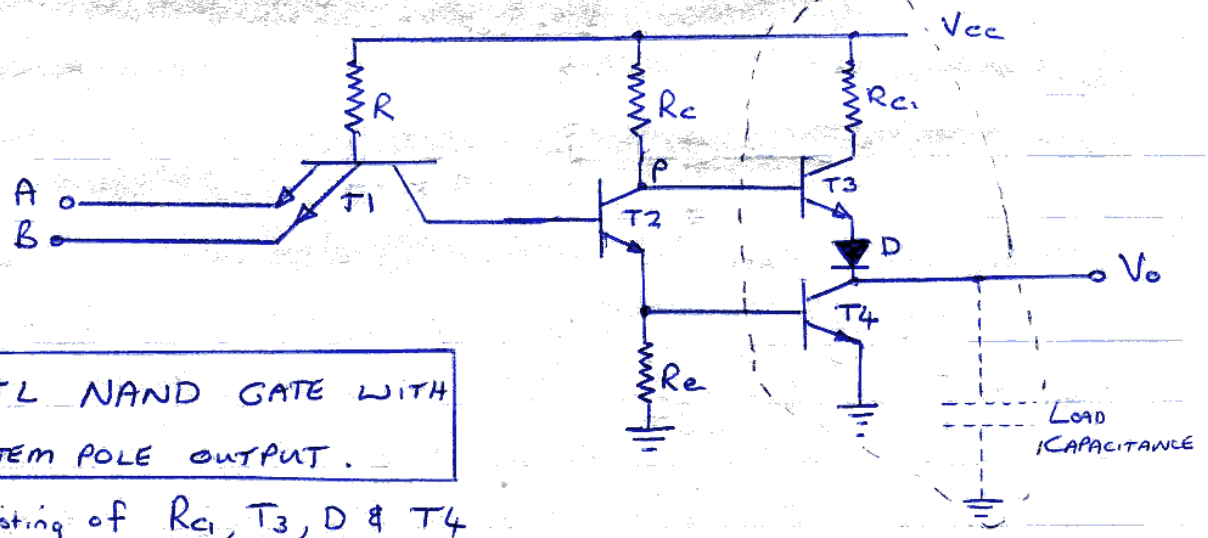


Consider the <u>dynamic</u> operation. Assume both inputs high, and then A is switched low. The output switches low to high in a time limited in DTL by the time it takes for the stored charge in the base of T2 to be removed by dissipation through Rb.

In TTL, however, this stored charge is removed by the multi-emitter transistor since during this switching transition it behaves as a common-emitter transistor in its active region, as the emitter is at a voltage near ground, thus providing current gain between base and collector which results in a large collector current which removes stored charge from the base of T2 and switches T2 off. Once the stored charge has been removed and T2 switches off, the behaviour of T1 reverts to that of a diode network. The switching speed is thus considerably improved.

12.5.3 TOTEM POLE OUTPUT STAGE

The switching speed of the TTL gate is still limited by the time required to charge a load capacitance through resistor R_c when the output goes from LOW to HIGH. This time constant can be decreased by decreasing R_c at the expense of increased power consumption. A significant speed improvement can be achieved by using a totem pole output stage, without much increase in power consumption.



TTL NAND GATE WITH TOTEM POLE OUTPUT.

consisting of R_{c1}, T_3, D & T_4

Operation:

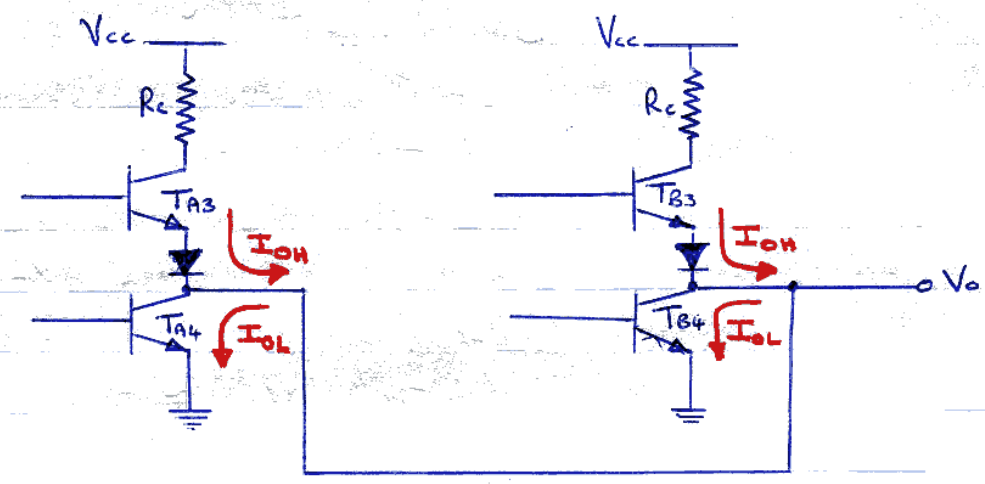
Both inputs high: T2 on, providing base drive to saturate T4. D prevents T3 being on, since voltage at P is $V_{BE_4} + V_{CE(SAT)_2} \simeq 0.95\,V$. As T4 is saturated the output voltage is about 0.2 V, the p.d between P and the output is about 0.75 V, which is the forward voltage drop across D. Thus $V_{BE_3} \simeq 0$ so T3; output LOW.
(normally enough to switch on T3)

One input goes low: T2 switches off, V_P rises and T3 switches on. At same time T4 switches off as its base drive is removed.

T3 provides active pull-up as it is in its active region and provides current gain, so the charging current from the emitter of T3 (which drives the load) is much greater than that provided by Rc.

Note that T4 will not switch off as fast as T2 since its stored charge can only leave through Rc. But T3 switches on as T2 switches off, raising the output before T4 is off. T3 must provide T4's collector current as well as charging the load capacitance; this does not severely effect switching speed due to T3's current gain. The switching speed can be improved if necessary by providing active pull-down for T4. T4 itself provides active pull down for the load capacitance since when both inputs go high again; T2 switches on, switching T4 on through its active region which provides current gain for discharging the load capacitance. Fanout is improved since the output current that can be sourced & sunk by T3 & T4 respectively is enhanced with respect to the previous stage.

2.5.4 Open Collector Output Stage



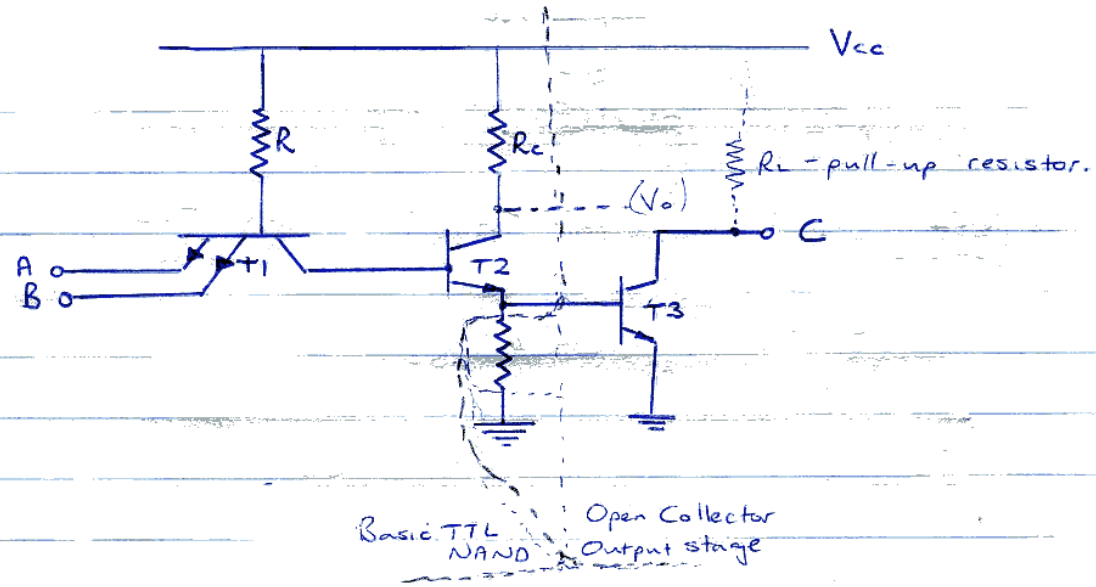
Connection of two totem-pole output stages

If we connect together the outputs of two TTL gates. If both gates are high, so is V_o, but if either gate (or both) is low V_o is low. So by connecting the outputs we are ANDing them without the need for an additional gate which is very useful.

Problem: Say A low, B high. Then $V_o \simeq 0.2$ V so T_{A4} is saturated so $V_{CE(B)} \simeq 0.2$ V.

Thus $I_{R_{C(B)}} \simeq \dfrac{5 - 0.2 - 0.2 - 0.75}{R_{C(B)}}$ as T_{B3} will be saturated due to its emitter voltage being pulled down by V_o going down to 0.2 V. Since R_c is typically 100 Ω this is about 38.5 mA. So T_{A4} must sink this current plus the current of any driven gates. But TTL gates are only rated for $I_{OL(max)}$ of 16 mA.

To overcome this, open collector gates are used.



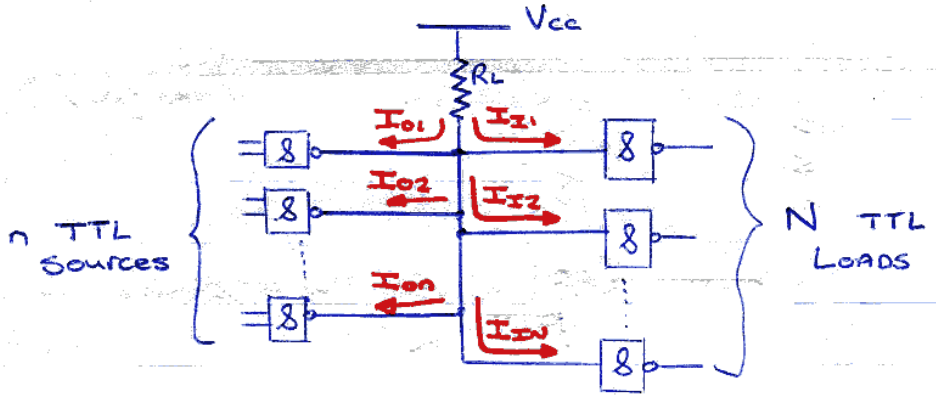
OPEN COLLECTOR TTL NAND GATE

To operate, C must be connected to Vcc through a pull-up resistor R_L – we have replaced the active pull-up of the totem-pole gate with a passive pull-up, causing an increase in switching times. However, a number of outputs can be wire-ANDed together, saving on gates and their associated propagation delays. Also useful for connecting several outputs to a common line bus. If all outputs are usually high, any single output going low will determine the state of the bus.

The value of R_L depends on the number of outputs being WIRE-AND-ed together and the number of loads being driven. It must be:

- large enough to prevent output transistors in the low state from being overloaded
- small enough to pull up high output voltages above the high threshold.

A range of values can be determined as follows:



INTERCONNECTION OF OPEN COLLECTOR GATES

Consider the case of a high output from the WIRE ANDed gates. Vo must be greater than V_{OH} (by defn) so

$$V_{RL} < V_{CC} - V_{OH}$$

So for any I_{RL}, we need: $I_{RL} \cdot R_L < V_{CC} - V_{OH}$

we can say then that: $R_{L(max)} = \dfrac{V_{CC} - V_{OH}}{I_{RL(max)}}$

From the figure we can see $I_{RL(max)} = \underbrace{n \cdot I_{OC}}_{\text{sum of currents sunk at outputs of driving gates.}} + \underbrace{N \cdot I_{IH}}_{\text{sum of inputs entering driven gates}}$

I_{OC} is the collector-emitter leakage current of an off transistor (the output transistor of the open-collector stage) given in the manufacturers specifications for the worst case high output voltage (V_{CC})

Thus $R_{L(max)} = \dfrac{V_{CC} - V_{OH}}{n \cdot I_{OC} + N \cdot I_{IH}}$ ⟶

To determine $R_{L(min)}$ we must find the smallest value of R_L that will protect the output transistor in the low output state. The worst case output current occurs when only one driving gate has a low output and the rest have high outputs. The low gate will have to sink through its output transistor the entire

current from R_L and all the input currents of driven gates (each at worst $-|I_{IL}|$)
The worst case output current of the gate is thus:

$$I_O = I_{RL} + N\,|I_{IL}|$$

The output current which the gate can handle and remain low is I_{OL}, so

$$I_{RL(max)} = I_{OL} - N\,|I_{IL}|$$

The smallest voltage across R_L is determined by the maximum specified output voltage V_{OL}, ie $V_{RL} = V_{CC} - V_{OL}$

so $$R_{L(min)} = \frac{V_{CC} - V_{OL}}{I_{OL} - N\,|I_{IL}|} \longrightarrow$$

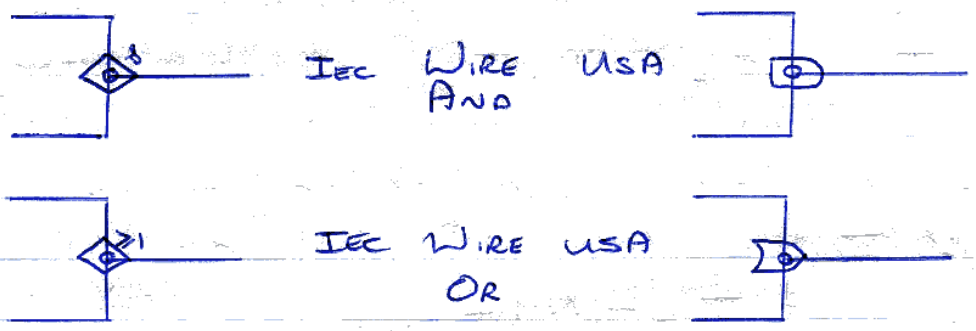
Fan out is limited by $N < -I_{OL}/I_{IL}$, since if $N = I_{OL}/I_{IL}$ we would require $R_L = \infty$. Also, as we increase N we reduce $R_{L(MAX)}$ and increase $R_{L(min)}$, giving a more stringent limitation on the fanout (ie $R_{L(MAX)} >= R_{L(min)}$) ie

$$\frac{V_{CC} - V_{OH}}{n I_{OC} + N I_{IH}} = \frac{V_{CC} - V_{OL}}{I_{OL} - N\,|I_L|}$$

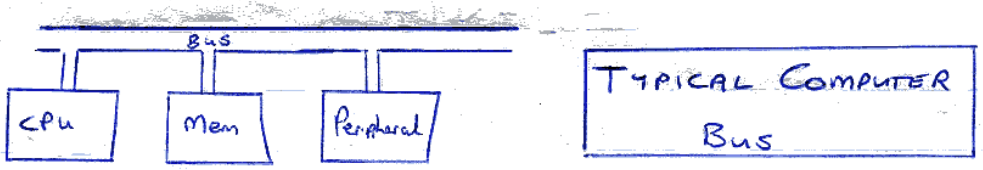
Solving this equation for N gives the fan out and hence the value of R_L needed for this maximum fanout.

Special buffers are manufactured with modified T3's having greatly increased I_{OL} and a much higher breakdown voltage, used to drive high voltage and/or high current loads such as lamps and relays.

Circuit Symbols for Wire-And and Wire-Or.



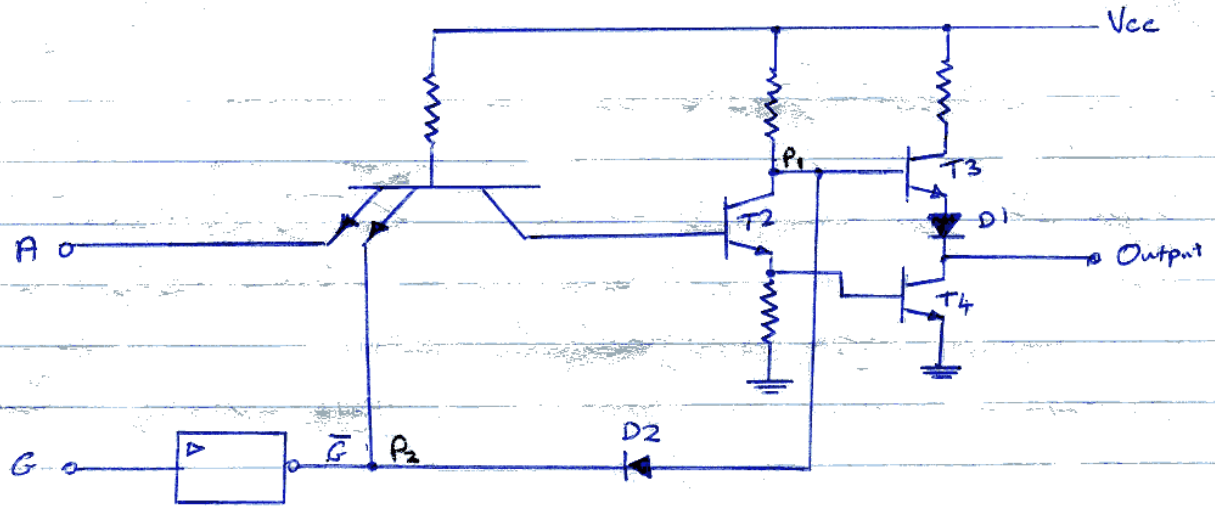
12.5.5 Tristate Buffers.



We need to communicate between any two devices on the bus. Open collector method has 2 disadvantages:

- speed (passive pull-up is slow)
- pull-up resistance must be carefully specified and recalculated each time a device is added/removed.

Instead, devices are connected to the bus through tristate buffers – inverters with a third high-impedance output state, the High-Z state. All devices except the two using the bus present a High-Z to the bus and thus do not load it.

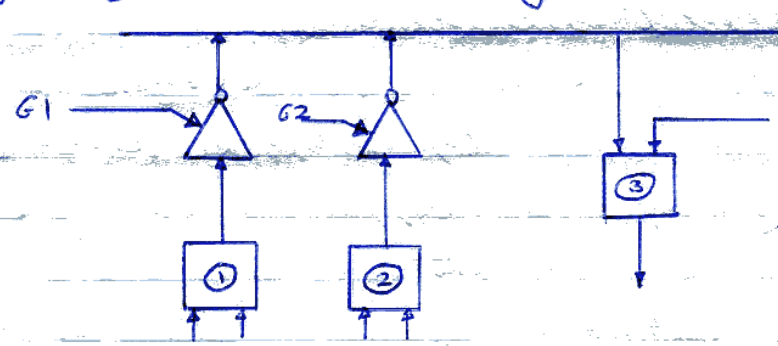


TRISTATE INVERTOR BUFFER

Operation: G high ⇒ $\bar{G}$ low ⇒ V_{BT1} low ⇒ T2 off ⇒ T4 off
In a normal NAND T3 would be on and output high but since $\bar{G}$ is low V_{BT3} is pulled low by forward biased D2 so T3 is off. So both output transistors are off so the circuit presents a high-impedance at the output

G low ⇒ $\bar{G}$ high and circuit behaves as a normal inverter: if A is high T3 is on ⇒ T4 on.
Since $\bar{G}$ high $V_{P2} > V_{P1}$ and $V_{P1} \simeq 0.95V$
so D2 is reverse biased, so $\bar{G}$ has no effect on T3 which is thus off giving low out
If A is low T2 is off ⇒ T4 off and T3 on ⇒ high out.

Eg 2 gates on a bus driving a third (note symbol)



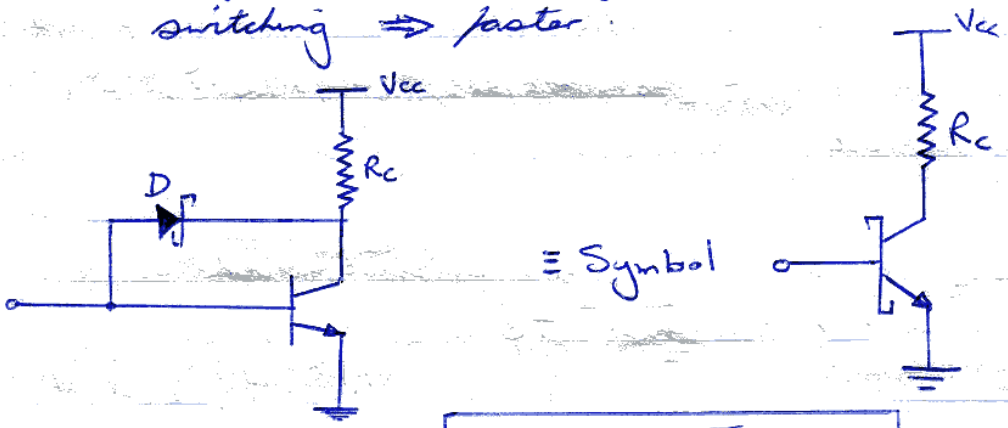
USE OF TRISTATE BUFFER IN A COMPUTER BUS

If gate 1 is to provide the input to gate 3 then G1 is low and G2 high

2.5.6 SCHOTTKY TTL

First consider Schottky diode , fabricated by making a junction between a metal such as aluminium and an n-type semiconductor. The metal acts as a p-type semiconductor (as it is a p-type impurity in sem-c's) and the Al-n-type junction behaves as a pn-junction except that:

- the barrier potential is much smaller than a normal pn junction, with a cut-in voltage of typically 0.35V
- there are no minority carriers on the metal side of the junction so storage time is negligible when switching ⇒ faster.



SCHOTTKY TRANSISTOR

The Schottky transistor is constructed by connecting a Schottky diode across the base-collector junction of an ordinary transistor. In the active region the transistor operation is unaffected since the B.C junction is reverse biased. In order to be saturated the collector potential must drop to about 0.55V below the base - this will never happen as at -0.35V the diode conducts. Thus the transistor can be driven to the edge of saturation (thus achieving the full switching effect) but prevented from gaining any saturation base charge ⇒ faster turnoff time.

Due to these superior switching speeds a TTL family constructed from Schottky transistors (with the same power requirements) has been developed, as well as a family called Low-Power Schottky which uses increased resistance values (same switching speed as TTL but lower power consumption).

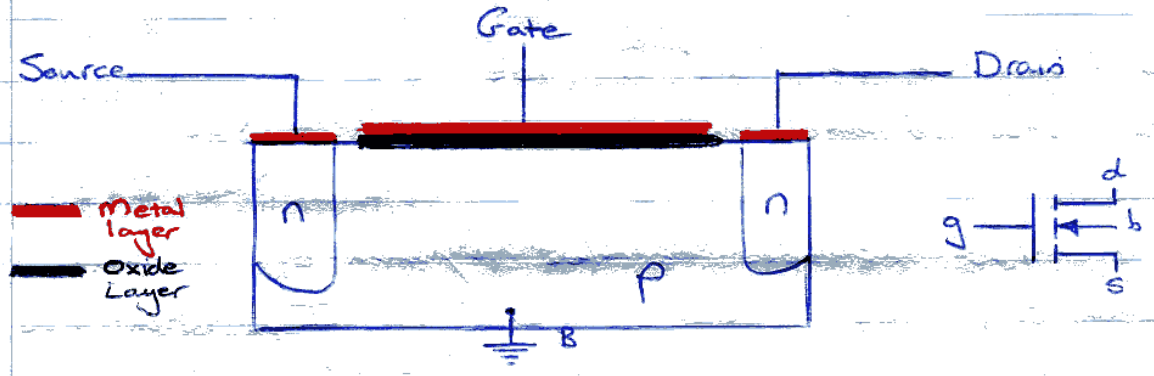
12.6 MOS AND CMOS (Metal oxide semiconductors)

TTL limitations:

- resistors occupy real estate and dissipate power
- bipolar transistors occupy excessive real estate as they must be isolated.

MOS technology, using MOSFETS, does not require the use of resistors and MOSFETS are self-isolating (slower than TTL)

12.6.1 MOSFETS. (MOS FIELD EFFECT TRANSISTORS)

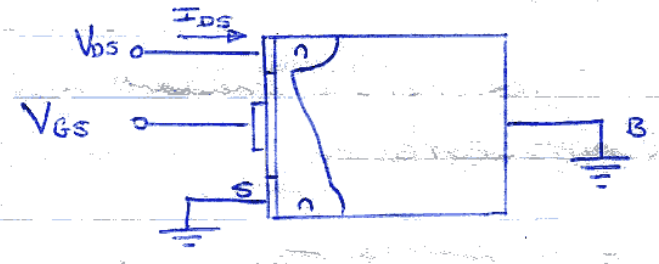


N-CHANNEL MOSFET

There is no conduction path between source and drain as one junction would always be reverse-biased. By making the gate (separated from the base by a thin metal oxide insulator)

positive with respect to the base, a field is created with electrons attracted to the gate forming an n-type channel between source and drain, making conduction possible. The arrow on the base connection (symbol) is in the direction of the base-channel p-n junction, indicating the MOSFET type (n- or p-channel).

The MOSFET described is called an "enhancement" MOSFET as it does not conduct until the gate voltage reaches ("is enhanced to") a threshold voltage V_T (between 2 and 4 V). An enhancement MOSFET has inherent noise immunity, unlike the depletion type.



OPERATING CONFIGURATION OF A MOSFET

Gate - base voltage at source is V_{GS}
Gate - base voltage at drain is $V_{GS} - V_{DS}$

So if V_{DS} increases to V_{GS}, "pinch off" occurs preventing any further ~~increase*~~ in I_{DS} with V_{DS} (dynamic equilibrium). The current will not collapse, as if this occurred the drain-source voltage would drop, thus reopening the channel. Unlike saturation in bipolar transistors, pinch-off in MOSFETS does not involve any stored charge, and so switch-off

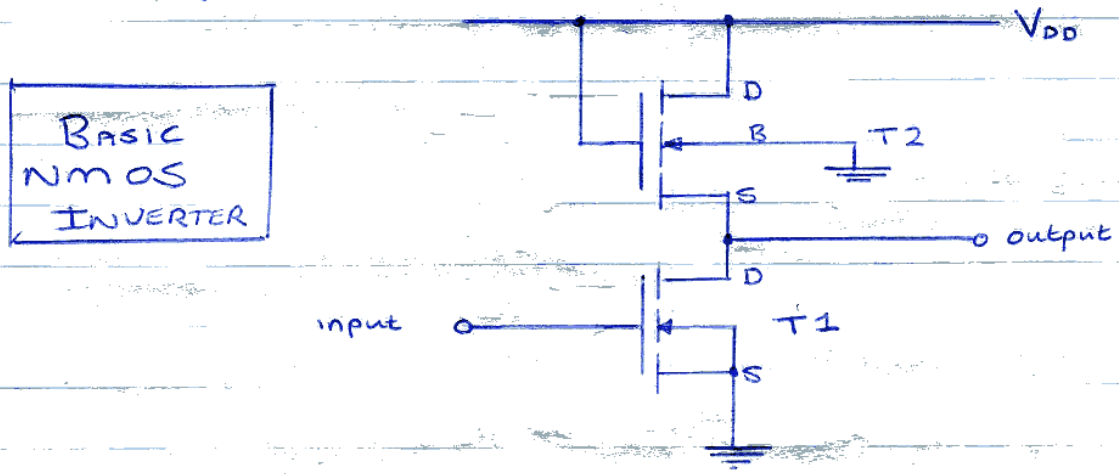
* I_{DS} continues at a constant level.

time is independent of whether punch off occurs or not.

PMOS are easier to fabricate, but NMOS is faster as electrons have about 3 times the mobility (and hence less resistance) than holes.

12.6.2 MOS LOGIC CIRCUITS

We consider NMOS (PMOS is identical in principle) switching

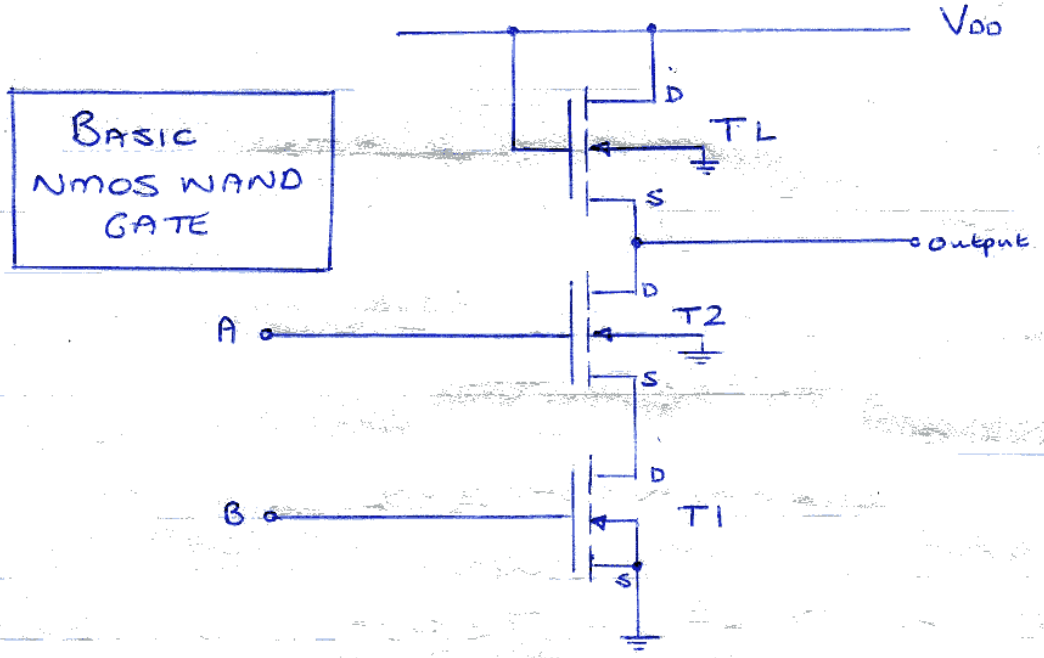


T2 is always on and acts as the load resistance for the switching transistor T1 (T2 occupies less real estate, easier to integrate than a passive resistor). All transistors are earthed at the base (the whole substrate forms all the bases and is earthed).

Operation: input low (below V_T) ⇒ T1 off ⇒ output high
input high ⇒ T1 on .. T1 & T2 are designed so that T1 has a lower channel resistance than T2. So output is from the potential divider of T1 & T2 and is low.

Input resistance is very high and thus draws negligible static current (0 picoamps). The loading is thus capacitive - high RC constants ⇒ slow. As integration densities improve (and T's get smaller) capacitance decreases. Can out not limited by static current loading but by dynamic switching time

Oxide layer easily ruptured by static discharge. Reverse - biased diodes are connected across the inputs protecting the MOSFETS by avalanche breakdown. - slightly increases circuit input capacitance

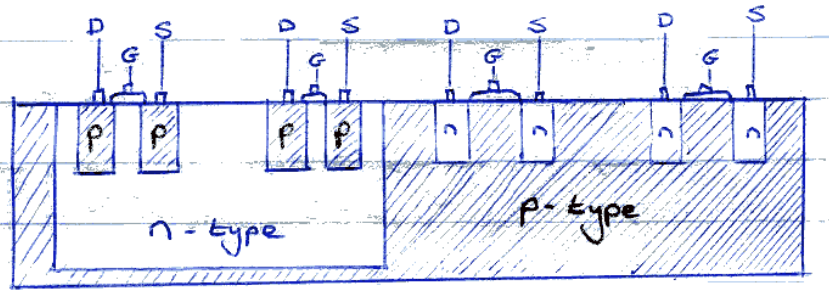


<u>Operation</u>: A, B both high ⇒ T1, T2 both on. T1 & T2 designed so their combined resistance is less than TL's so output is low

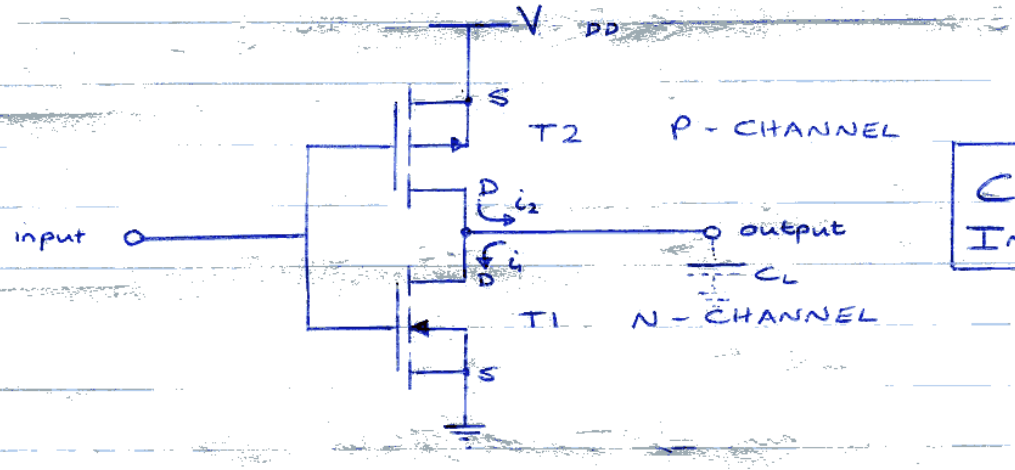
Either/both A, B low ⇒ either/both T1/T2 off ⇒ output pulled high.

12.6.3 CMOS LOGIC CIRCUITS.

Faster switching and reduced power consumption over NMOS/PMOS. CMOS is more expensive as both N- and P-channel MOSFET's must be integrated onto the same piece of silicon



X - SECTION OF CMOS I.C.



CMOS INVERTER

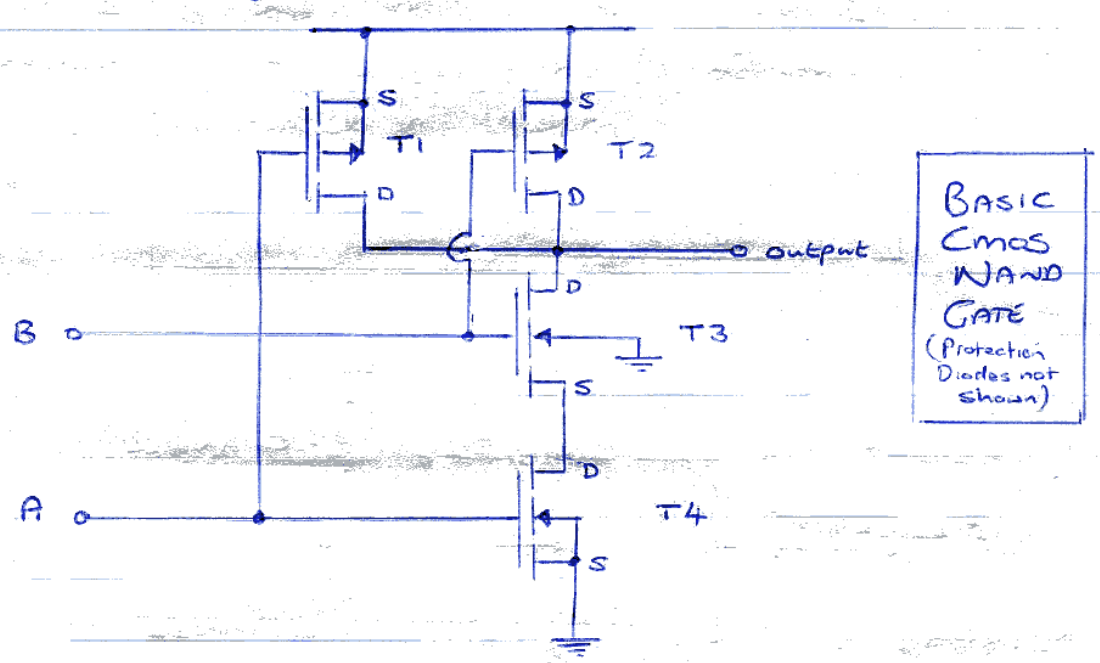
A symmetrical pair of N- channel and P-channel MOSFET's are joined by their drains. In each case the base is connected to the source of the transistor which is connected to one of the power rails.

Operation: input low ($V_{in} < V_{DD} - V_T$) ⇒ T1 off, T2 on ⇒ output high
input high ⇒ T1 on ($V_{in} > V_T$), T2 off ($V_{in} > V_{DD} - V_T$) ⇒ output low.

Resistance in CMOS is much less than NMOS/PMOS and transistor switching is faster: output goes high ⇒ switching time is $R_{T2} \cdot C_L$ (load cap.) and $R_{T2} \simeq R_{T1}$; output goes low ⇒ s.t. determined by current drawn from C_L

There is no static power dissipation (due to capacitive loading). The power dissipation is inversely proportional to the switching time.

A wide range of power supply voltages is possible; not the case for any other family.

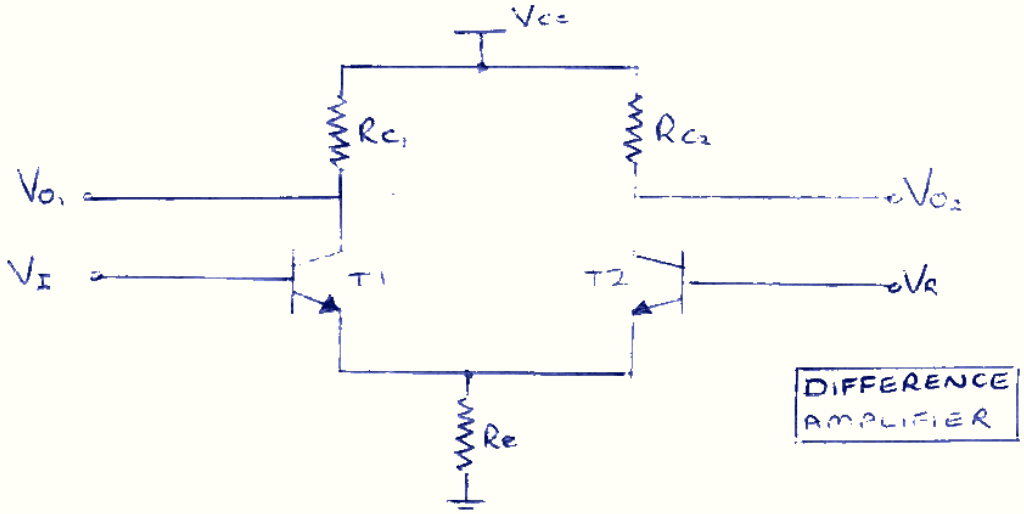


Operation: A high ⇒ T1 off, T4 on } ⇒ output low
B high ⇒ T2 off, T3 on

A low ⇒ T1 on, T4 off ⇒ output high
B low ⇒ T2 on, T3 off ⇒ output high.

12.3 EMITTER COUPLED LOGIC (ECL)

The main problem influencing the speed of bipolar logic was the removal of base charge from the saturated transistor. It would be much better if we could use the transistors purely in their active region. The difficulties with this are that the range of V_{BE}'s for which a transistor is in its active state is very small and highly influenced by temperature drift, tolerances, etc. ECL logic is based on the difference amplifier, an alternative switching mode of the transistor.



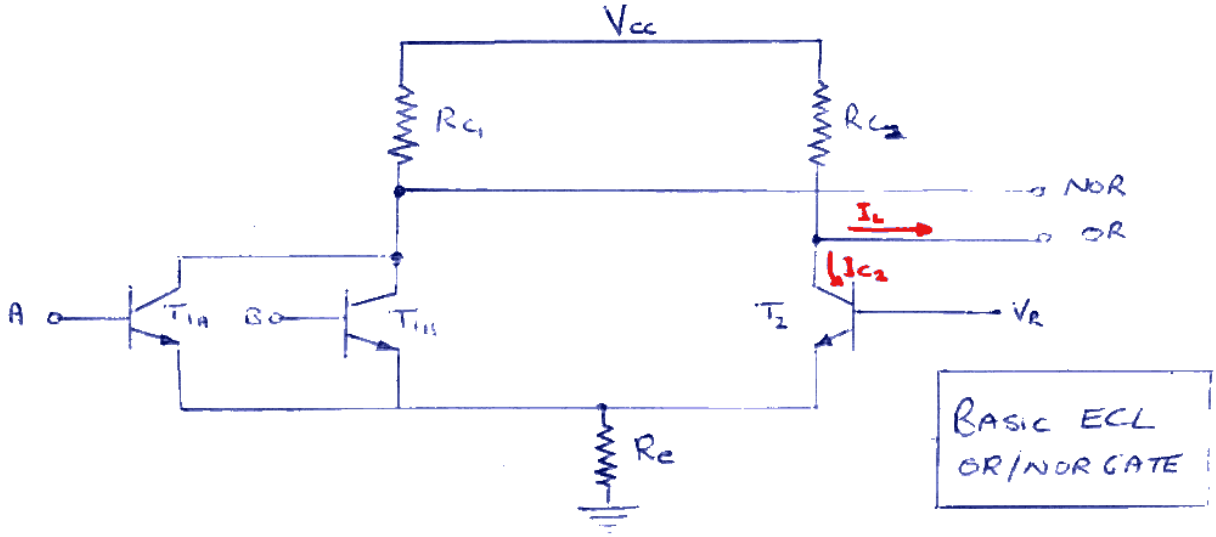
Base of T2 is held at a reference voltage V_R (which can be made temperature stable). If T1 is off, T2 is then on with collector current $I_e \approx \frac{V_R - V_{BE}}{R_E}$. If V_I goes above V_R by a small amount δ, the current is effectively switched but is still $I_e \approx \frac{V_R + \delta - V_{BE}}{R_E}$

Thus the "same" current is switched from T2 to T1 with only a small input transition

Operation: V_I low $\Rightarrow T_2$ on, R_{C2} and V_{O2} low

V_I high $\Rightarrow T_1$ on, T_2 off, R_{C2} has no current, pulling V_{O2} high. <u>NB</u> V_{O2} value is independent of V_I

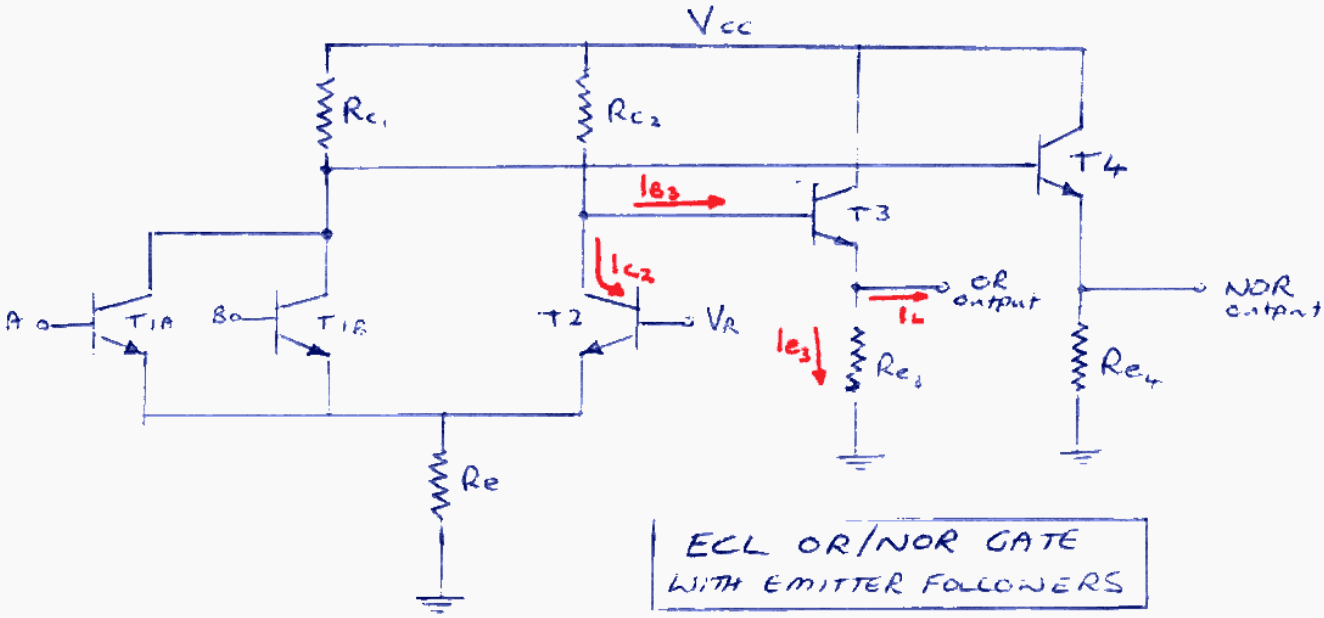
Switching transition of V_I is small (~50mV) and $R_{C1} \simeq R_{C2}$ so V_{O1} (low) $\simeq V_{O2}$ (low)
Increasing V_I causes V_{O1} to decrease - as V_{O1} is low this is not a problem, except if T_1 is saturated (slows down)



Operation:

A	B	T2	V_{C1}	V_{O2}
L	L	On	H	L
H	L	Off	L	H
L	H	Off	L	H
H	H	Off	L	H

This is an oversimplification - in fact the outputs are connected to emitter followers.



ECL OR/NOR GATE
WITH EMITTER FOLLOWERS

- Shift the level of the output signal down by V_{BE} of an ON transistor, matching output to input voltages
- Provide low output impedance, improving fanout & speed.
- Allow WIRE-OR'ing of outputs (see below)

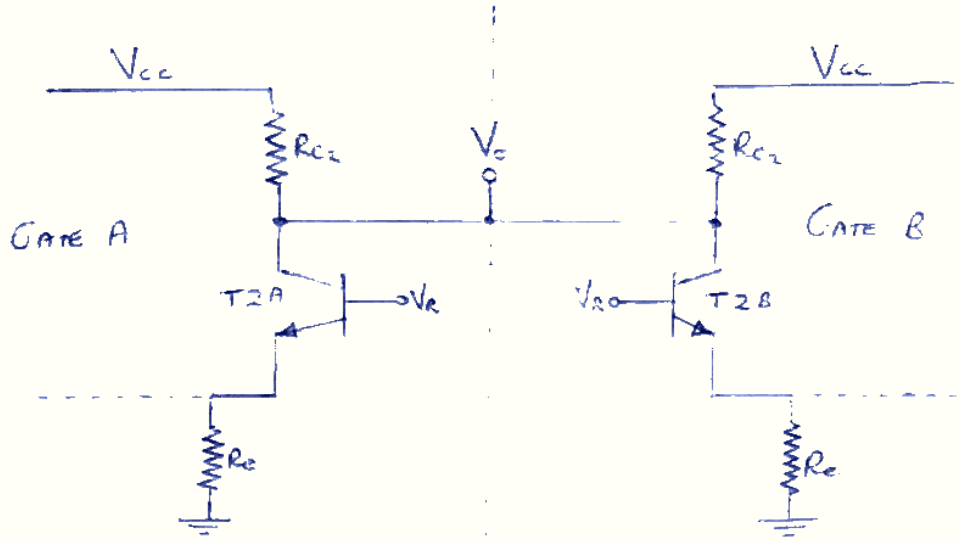
Consider the relationship between output load current & output voltage (for simplicity consider OR gate only). Without the emitter follower the output voltage is $V_{O2} = V_{CC} - (i_{C2} + i_L) R_{C2}$. With the emitter follower, the output voltage is

$$V_{O2} = V_{CC} - (i_{C2} + i_{B3}) R_{C2} - V_{BE}$$

where $i_{B3} = \dfrac{i_{e3} + i_L}{\beta_B + 1}$, $\beta \simeq 100$ ie $i_{B3} \simeq \dfrac{i_{e3}}{\beta} + \dfrac{i_L}{\beta}$

So $V_{O2} = V_{CC} - V_{BE} - (i_{C2} + i_L/\beta + i_{e3}/\beta) R_{C2}$

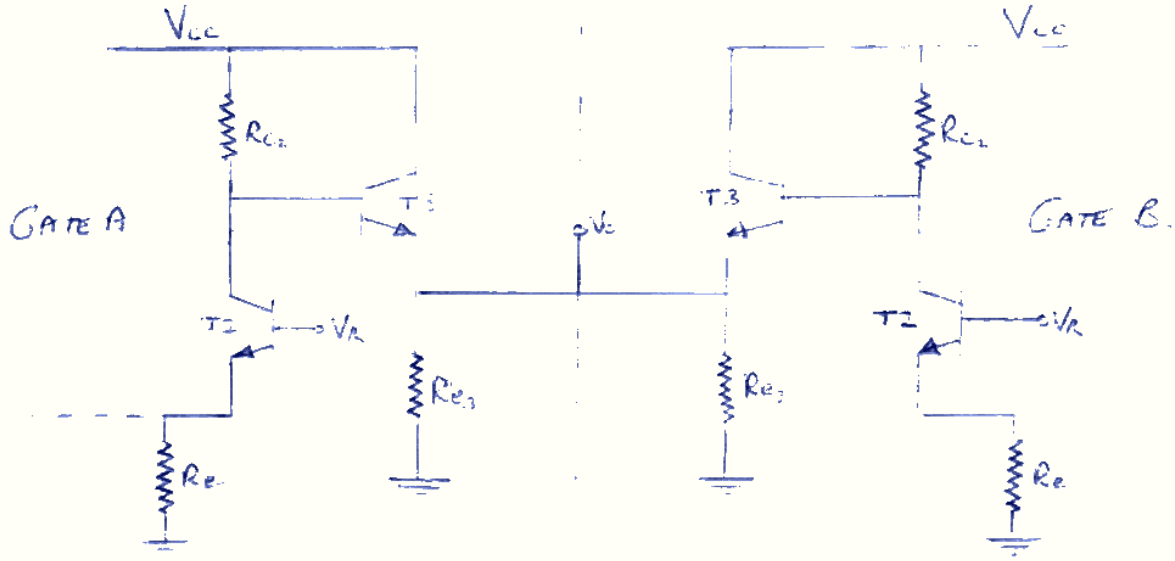
Thus the dependence of the output voltage on the load current is reduced by a factor of β



GATES WITHOUT EMITTER FOLLOWERS

WIRE-OR

GATES WITH EMITTER FOLLOWERS



In the first case, if one output is high and the other is low then the output [illegible] — or low effectively had since it is loaded by two equal resistances in parallel. However, the collector current does not change since this is determined by the emitter current which is dependent only on VR and Re. Thus the output voltage is half way between H & L - no good.

In the second case, let gate A be high (T_{2A} off) and gate B be low (T_{2B} on). Both T3's are on so the current through each is increased as the emitter resistor is now approximately R_{e3} of A in parallel with R_{e3} of B. The output is determined by the load resistance/β so is hardly affected, so T_{3A} pulls the output high.

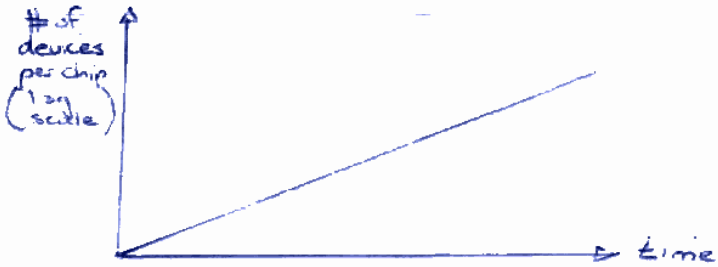
Fanout. Low state: to drive the base of an off transistor uses minimal current drain. Current drawn from the output lowers the output voltage so there is an infinite low state fanout.

High state: to drive the base of an active transistor uses low current drain. The emitter follower has a high current driving capability. The fanout is thus limited by switching speed.

Note:
- ECL interconnections behave as transmission lines
- the reference voltage is supplied by an active circuit with the same temperature drift characteristics as transistors, avoiding the requirement of an additional power rail
- ECL has a low noise margin and is thus very susceptible to temperature drift - hence the active reference voltage.

13 APPLICATION SPECIFIC INTEGRATED CIRCUITS (ASICs)

Moore's Law.



Logic families usually have a series of standard chips to achieve cost effectiveness of large quantities. However, most designs require "random" logic - implemented using SSI and MSI chips ("TTL glue" USA term). This causes increased board space and heat dissipation, and loss of reliability.

A major cost factor in IC manufacture is design time (i.e. circuit design, layout checking, mask production, manufacture with 8 or 9 mask layers)

Now CAD techniques speed-up / improve design process so the design content of a specific IC is becoming a smaller fraction of the overall cost ⇒ ASIC's.

13.1 CUSTOM IC's. (complete design needing all mask layers)

13.2 SEMI-CUSTOM IC's ("USA term silicon compilation")

These use fully designed standard cells (eg µP, memory, I/O controllers) stored on a computer, put together with a few gates. All mask layers still need to be manufactured.

13.3 GATE ARRAYS

These are arrays of standard logic gates. Application specific design is achieved by choosing interconnections so only the metal layer mask needs to be developed specifically; all other mask layers are standard for all chips.

13.4 FIELD-PROGRAMMABLE LOGIC.

The allow more complexity than gate arrays. Customisation does not require alterations to the manufacturing process or expensive CAD facilities.

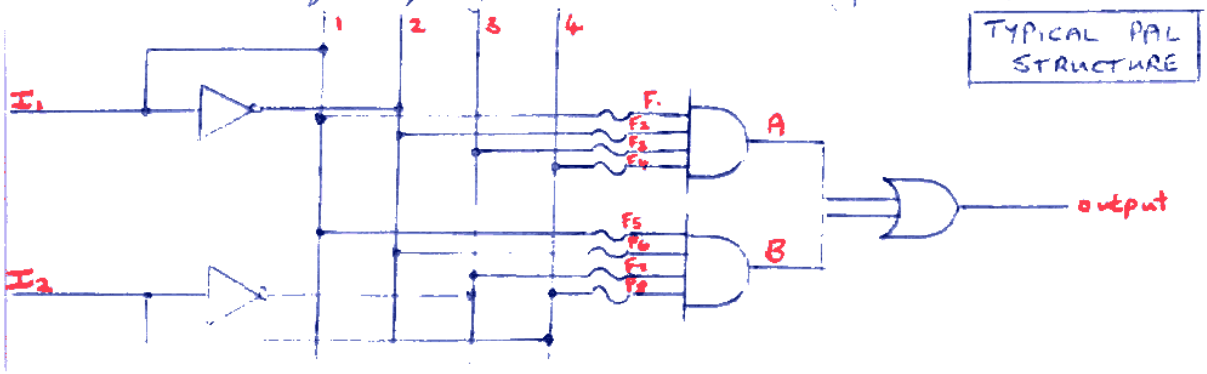
PROMs – small fuses which can be blown by applying a high voltage (typ ~20V).

EPROMs – CMOS chips with gates which can retain their charge until exposed to UV light.

PALS – blow fuses to achieve required logic functions from a standard array of gates. The fuses determine the interconnection between gates.

13.4.1 PROGRAMMABLE ARRAY LOGIC.

These consist of a programmable AND array, with a fixed OR array.



The output of this gate is A+B. We can regard the input to an AND gate (say A) as

$A = (I_1 + F_1) \cdot (\bar{I}_1 + F_2) \cdot (I_2 + F_3) \cdot (\bar{I}_2 + F_4)$

where a fuse has value 0 if intact, 1 if blown (ie inputs whose fuses are blown are 'high')

The PAL consists of two parts:

PROGRAMMABLE AND ARRAY – current inputs and their inverses are fed through fuses into multiple input AND gates

FIXED OR ARRAY – the outputs of the AND gates are fed into OR gates in a fixed (ie non-fused) way.

PAL NOTATION

Inputs are fed into inverters and connected to vertical input lines which cross horizontal product lines with the AND gates. Fuses are located at the intersection of input and product lines. If a fuse is <u>intact</u> the intersection is marked with an X otherwise it is unmarked. Product lines are analogous to a WIRE AND

Fuses are defined by the input and product lines at the intersection of which they lie. Input lines can be current inputs, their complements, or feedback outputs.

EXAMPLES OF PAL FUSE PATTERNS

The first example we consider is a Full Subtractor. We subtract X from Y with a borrow B, giving result D with borrow C. The truth table and expressions are

X	Y	B	D	C
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

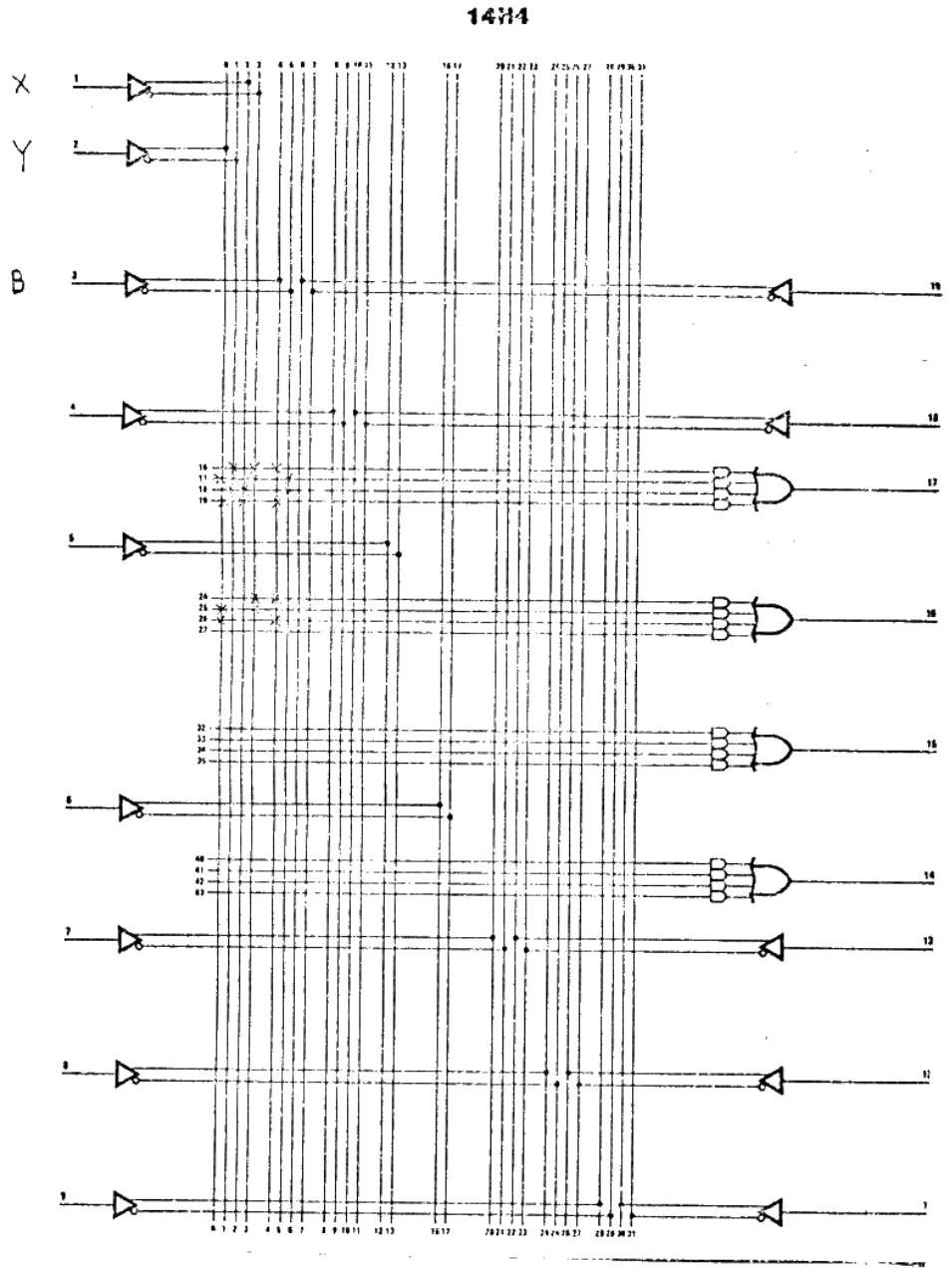
TRUTH TABLE & FUNCTIONS FOR A FULL ADDER

$$D = \bar{X}\bar{Y}B + \bar{X}Y\bar{B} + X\bar{Y}\bar{B} + XYB$$

$$C = \bar{X}B + \bar{X}Y + YB$$

PAL FUSE PATTERN FOR A FULL ADDER.

Fig 5.5.



4-Bit Gray to Binary Converter

G_4	G_3	G_2	G_1	B_4	B_3	B_2	B_1
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
1	0	0	0	1	1	1	1
1	0	0	1	1	1	1	0
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	0	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	0	1	0	1	1
1	1	1	1	1	0	1	0

$B_4 = G_4$

$B_3 = \overline{G_4} G_3 + G_4 \overline{G_3}$

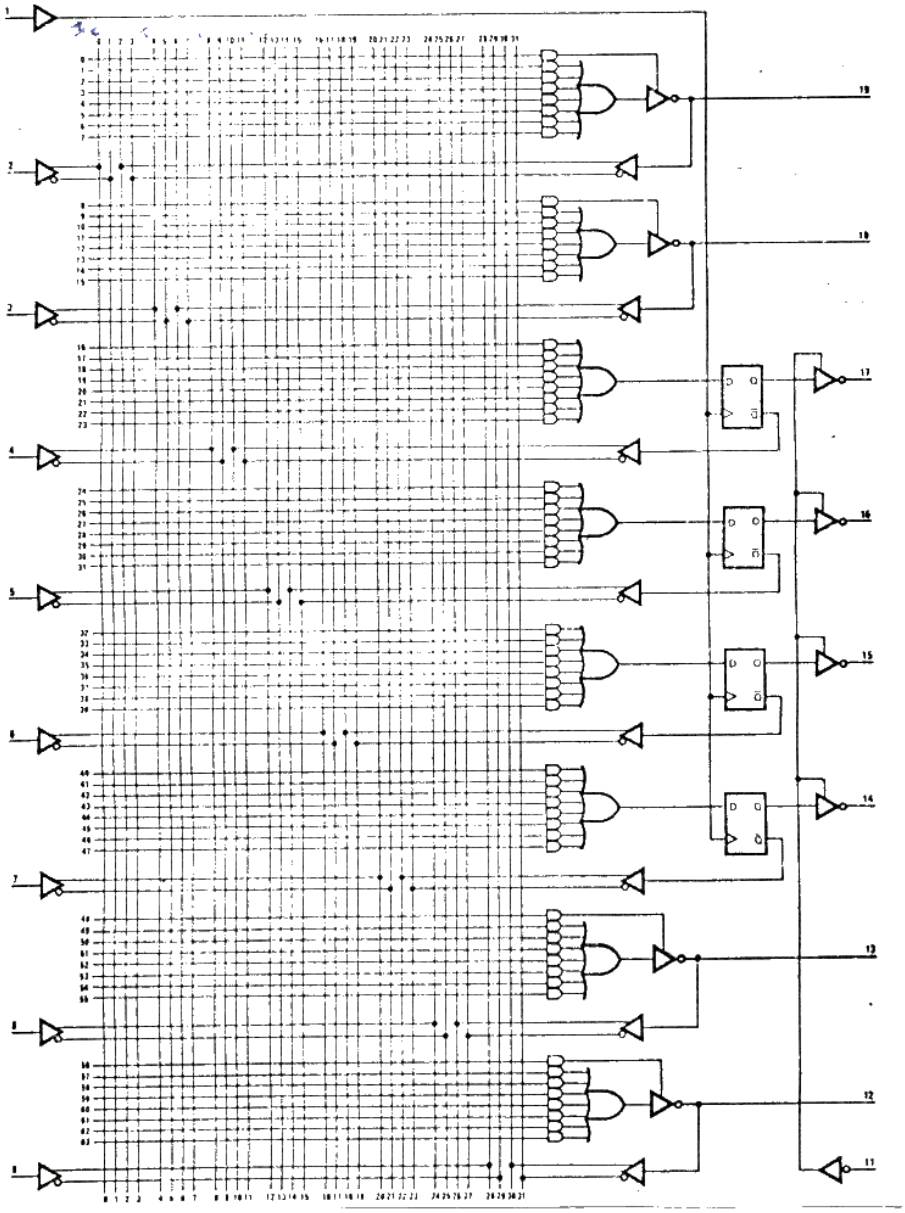
$B_2 = \overline{G_4}\,\overline{G_3} G_2 + \overline{G_4} G_3 \overline{G_2} + G_4 G_3 G_2 + G_4 \overline{G_3}\,\overline{G_2}$

$B_1 = \Sigma$ product terms

A PAL could easily be used for this conversion

13 + 2 PROGRAMMABLE I/O

16R4



Pin 19 of the above figure can be either an input or an output, since the output of the AND-OR array is fed via a tristate buffer to pin 19. The tristate buffer is enabled / disabled by the function of product line 0. If 0 is high the buffer is enabled, pin 19 is an output and the output is fed back to input lines 2 and 3. If 0 is low, the buffer is disabled, pin 19 is a dedicated input feeding input lines 2 and 3

The product line C can be

① a function of the inputs

② permanently high (fuse blown)

③ permanently low (all fuses intact)

1343 PALS WITH REGISTERED OUTPUTS WITH FEEDBACK

The OR outputs are sent to the D input of edge-triggered D type flip flops. The Q output is fed via a tristate buffer to the output pin as well as being fed back to the input lines (and its complement) of the AND OR array (feedback ⇒ synchronous state machine design)

Example MOD-8 GREY CODE COUNTER

PRESENT STATE			NEXT STATE		
A	B	C	D_A	D_B	D_C
0	0	0	0	0	1
0	0	1	0	1	1
0	1	0	1	1	0
0	1	1	0	1	0
1	0	0	0	0	0
1	0	1	1	0	0
1	1	0	1	1	1
1	1	1	1	0	1

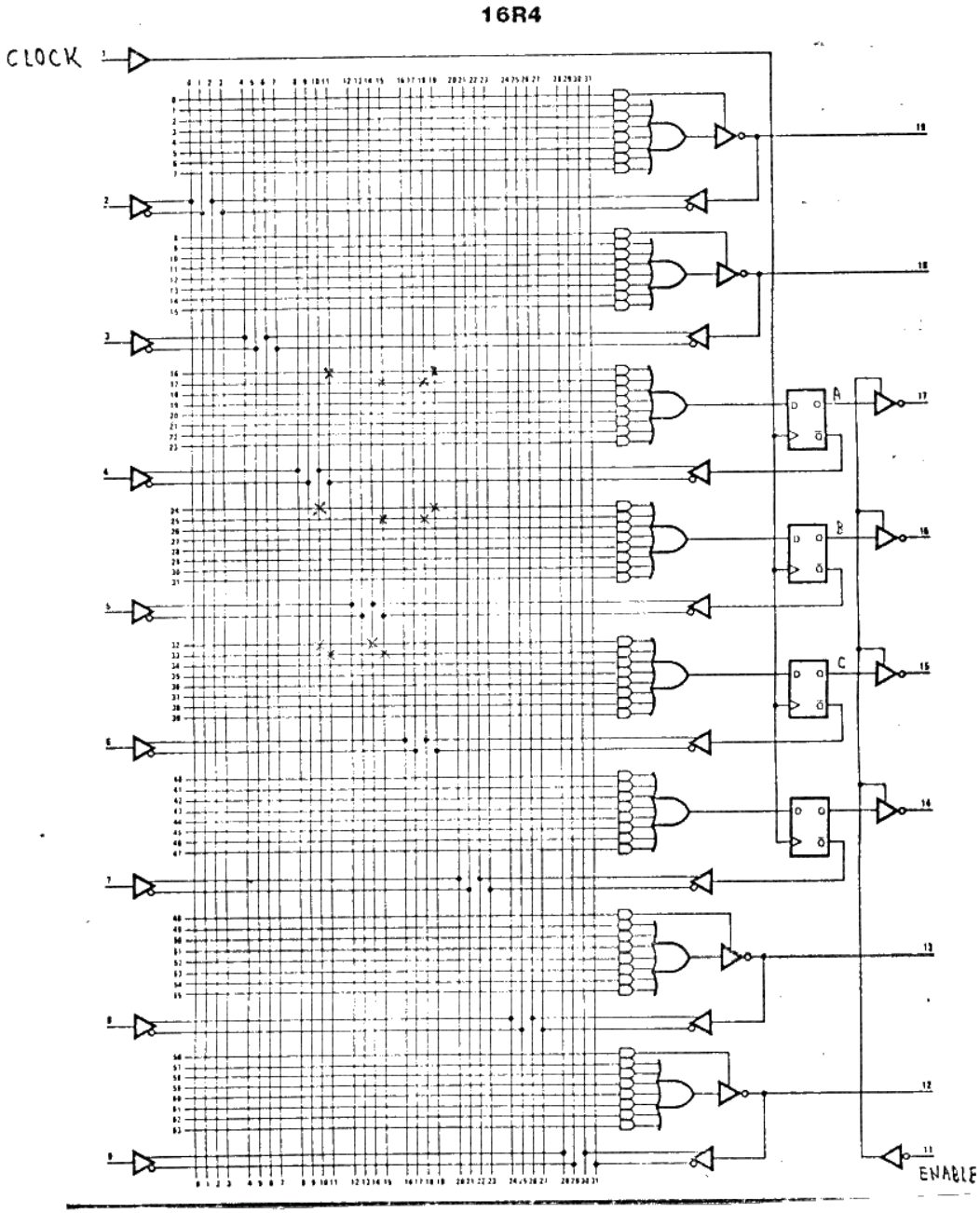
$D_A = AC + B\bar{C}$

$D_B = \bar{A}C + B\bar{C}$

$D_C = \bar{A}\bar{B} + AB = \overline{A \oplus B}$

see opposite for bit pattern.

10	11	14	15	19	11	15	18	10	19	13	18	10	14	11	15
$\bar{A}$	C			A	C	B	$\bar{C}$	$\bar{A}$	C	B	$\bar{C}$	$\bar{A}$	$\bar{B}$	A	B



13.4.4 PALS WITH EXCLUSIVE - OR GATES

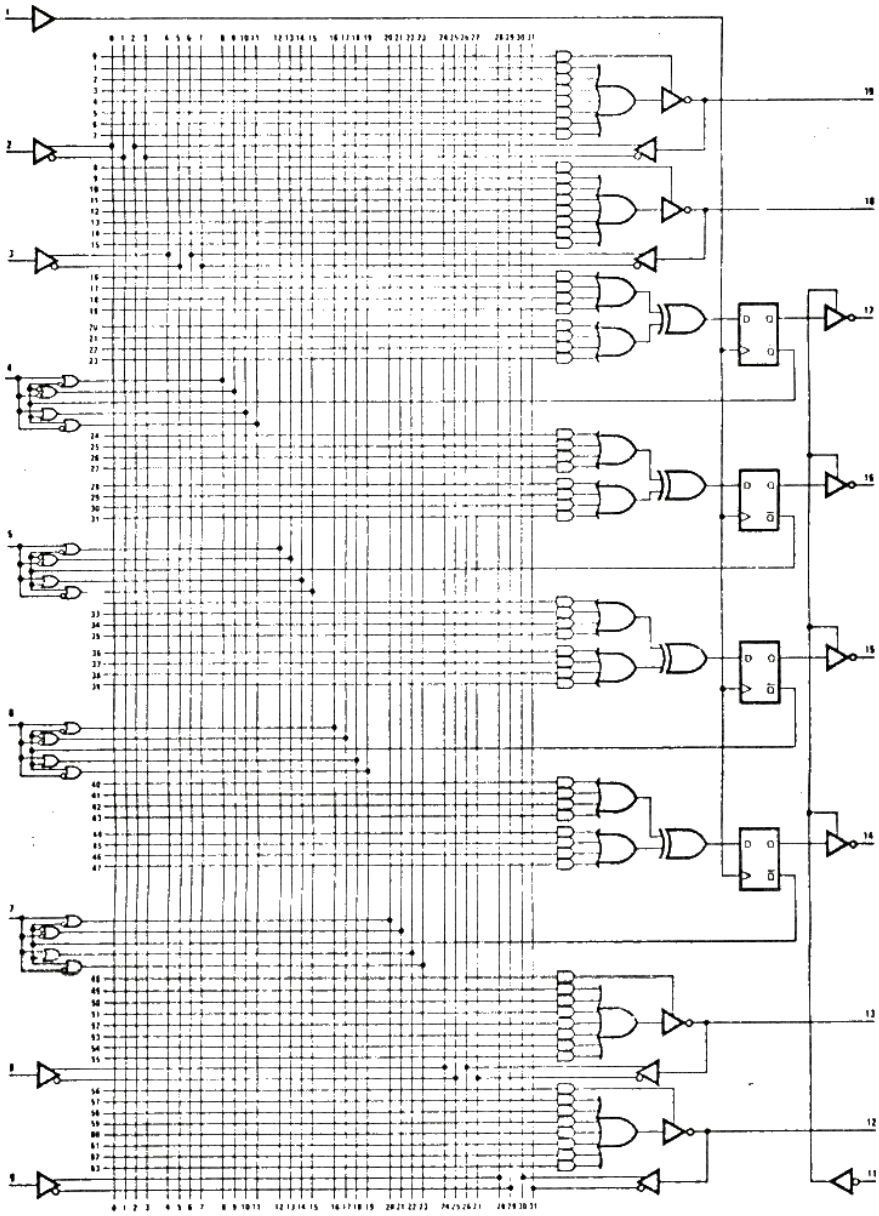
Arithmetic circuitry uses XOR / XNOR gates so PALS have been designed which provide these functions at both the input and the output of the AND-OR array (see ~~[illegible]~~ for diagram)

13.4.5 PAL DESIGN SOFTWARE

CAD design tools are available for PALS eg PALASM

which takes a boolean function and produces the fuse map. This fuse map is used by a PAL programmer which selects a fuse via a specified pattern on the I/O pins. By placing a high voltage on some of the pins, the fuse is blown. The programmer checks that the fuse is blown and repeats the attempt if it was unsuccessful.

16X4



14 ANALOGUE / DIGITAL & DIGITAL / ANALOG CONVERTORS

14 If we are to use digital circuits to process analogue signals, we must be able to convert between the two forms. A typical application would be computer controlled heating: an electrical thermometer feeds its output to an A/D convertor, the output of which is processed and fed to a D/A convertor controlling a heater element. It is important to note that the conversions and calculations take time, so it is not possible to continuously vary the heater; rather we are effectively digitising time by breaking it up into discrete units in which we sample the input. The Shannon-Nyquist theorem states that as long as we sample at a rate of more than twice the highest analogue frequency in the system, we will not be losing any information due to the digitising of time

14.2 D/A CONVERTORS

A digital number in binary representation is

$$B = J_{N-1}\, J_{N-2} \ldots\ldots\ldots J_1 J_0$$

where J_k is the representation of the k'th bit in the N-bit number. The decimal equivalent is

$$D = J_{N-1} \cdot 2^{N-1} + J_{N-2} \cdot 2^{N-2} + \ldots + J_1 \cdot 2 + J_0$$

To convert this number to an analogue value we have to represent it as a series of discrete analogue voltages where each analogue voltage differs from the next one by a fixed voltage, the step size or resolution of the convertor. If a convertor has a step size of δV then one decimal unit is represented by an analogue change of δV. So the analogue output of the convertor is given by:

$V_A = D \cdot \delta V$

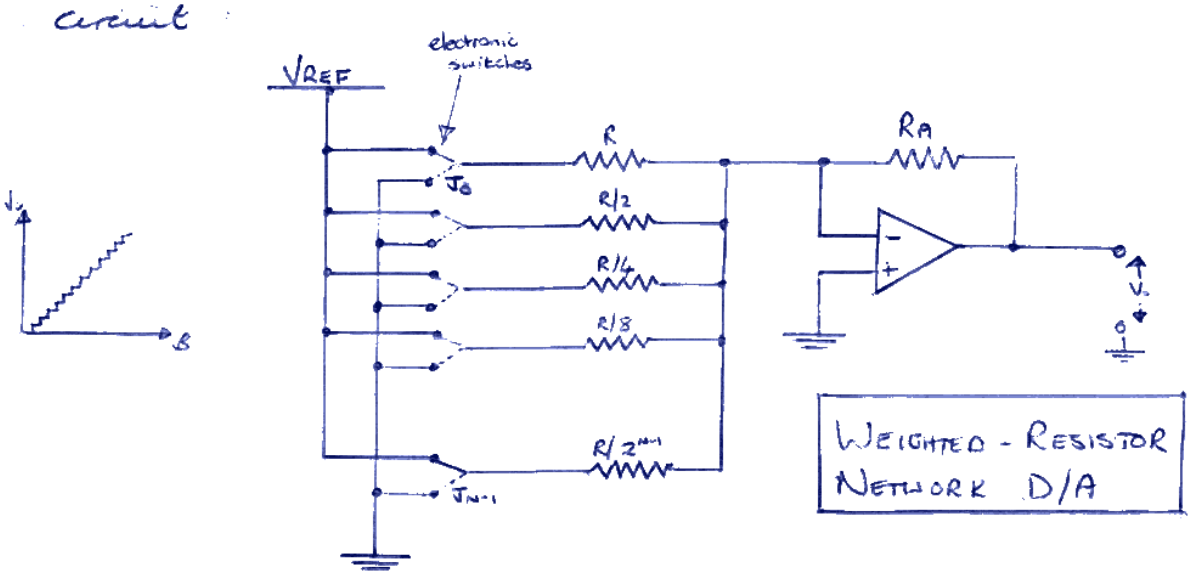
so

$$V_A = J_{N-1}(2^{N-1}\delta V) + \dots + J_0 \cdot (2^0 \cdot \delta V)$$

The basis of D/A converter is thus that each increment in decimal value should increment the analogue output of δV

14 2.1 <u>WEIGHTED - RESISTOR NETWORK D/A.</u>

The simplest D/A converter is made by using a weighted-resistor network and an operational amplifier adder circuit:



The electronic switches are single pole double throw switches whose position is controlled by a logic signal. These switches are set to reflect the bit pattern of the input digital number, each bit controlling one switch, the k'th msb controlling the switch with resistance $R/2^k$.

When a bit is set to 1 and the resistor connected to the supply voltage the current that flows in that leg of the network is given by:

$$i_k = V_{ref} / (R/2^k) = \frac{V_{ref} \cdot 2^k}{R}$$

so in general the current in any leg of the weighted resistor network is given by:

$$i_k = J_k \cdot V_{ref} \cdot 2^k / R \quad (\text{since if } J_k = 0,\ i_k = 0)$$

Since the currents in all legs of the weighted-resistor network are summed through the operational amplifier feedback resistance R_A, the output voltage V_o is given by:

$$V_o = -R_A \left[\frac{J_{N-1} V_{ref} 2^{N-1}}{R} + \dots + \frac{J_0 V_{ref} 2^0}{R} \right] = \frac{-R_A V_{ref}}{R} \left[\dots \right] = \frac{-R_A V_{ref} D}{R}$$

and thus $\delta V = \dfrac{-R_A \cdot V_{ref}}{R}$

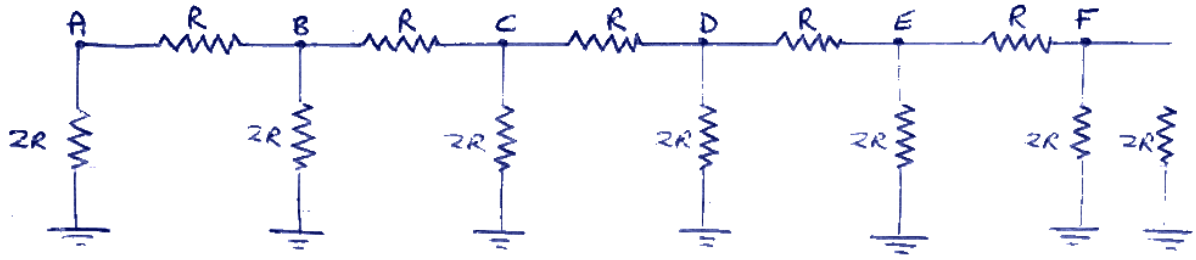
Note that to have a positive step size we need a negative supply voltage. The output of a D/A converter is called a staircase ramp.

The weighted-resistor network has poor accuracy and linearity as these are critically dependent on the resistance values in the network, and we are dealing with a large range of resistances. There are disadvantages as far as connecting switches between V_{ref} and resistance go as well:

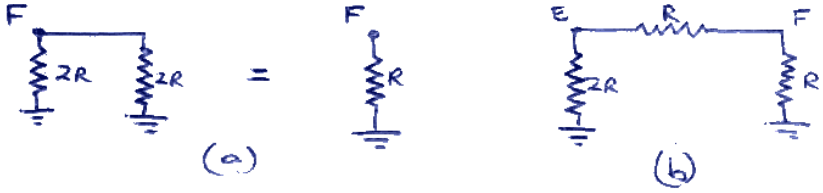
- current in leg is either 0 or i_k, hence when input changes current and hence output must settle.
- voltage across open circuit of switch is full supply voltage - switch must be designed not to break down
- load resistance presented to supply varies depending on input, between R and $R/2^{N-1}$, so unless supply impedance is zero, inaccurate result.

14.2.2 R-2R LADDER D/A CONVERTOR

The so-called R-2R resistance ladder shown below is a way of achieving the same currents at the summing input of an operational amplifier as for the weighted resistor network, but requiring only two resistance values, R and 2R.

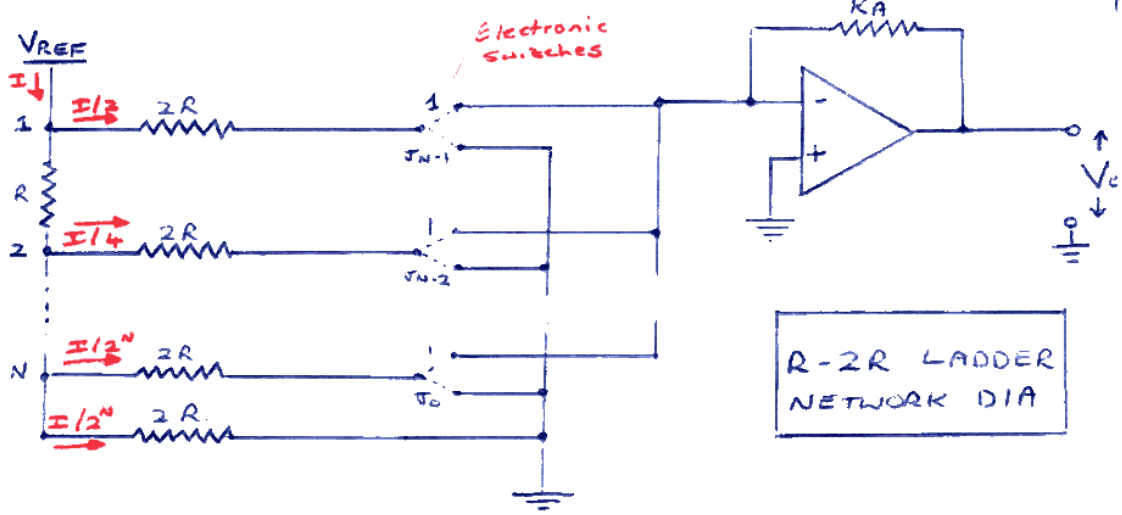


To understand the operation of the R-2R ladder consider the resistance between node F and ground. As shown below (a) this is R. Thus the resistance between node E of the network and ground as viewed from the left of node E is also R, since between node E and ground we have two equal resistance paths of 2R each as shown in (b).



Similarly, the resistance between any other node and ground "looking right" is also R.

At each node in the network current i flowing into the node from the left will divide into equal currents i/2 flowing out of the node. Thus the resistance presented to a supply voltage connected to node A is R, and each earthed leg of the R-2R ladder has a current equal to half of that of the leg to the left of it. This can be used as the basis of a D/A convertor based on a summing operational amplifier:



Again as in the weighted resistor network D/A each leg has an electronic switch which is set by the value of the corresponding bit in the binary number. The current in a leg is independent of the switch setting (as the input to the op-amp is a virtual earth). VREF has a constant load resistance R, and supplies a current $I = V_{REF}/R$ into the network. The current in the leg corresponding to bit J_K of the input is $I/2^{N-K}$. Thus the analogue output is

$$V_o = -R_A\left[\frac{J_{N-1}\,I}{2^1} + \frac{J_{N-2}\,I}{2^2} + \cdots + \frac{J_0\,I}{2^N}\right] = -R_A\,I\sum_{i=1}^{N}\frac{J_{N-i}}{2^i}$$

$$= -\frac{R_A\,I}{2^N}\sum_{i=1}^{N} J_{N-i}\cdot 2^{N-i} = \frac{-R_A\,I\,D}{2^N} = \left[\frac{-R_A\,V_{REF}}{2^N\,R}\right]D$$

so $$\delta V = -\frac{R_A\,V_{REF}}{R\cdot 2^N}$$

The position of the switches in the ladder introduces further advantages over the weighted resistor network

- the load resistance presented to VREF is constant, whereas a WRN has load resistance varying from R to $R/(2^N-1)$, which affects the actual potential presented the converter, causing non-linearity and other errors.

- leg currents are constant and independent of switch settings, whereas in WRN it is either 0 or V_{REF}/R_K (where R_K is the leg resistance). Thus the WRN has greater 'settling time' than the R-2RL.
- the open circuit potential across switches in R-2R is approximately zero, whereas in WRN it is the full supply voltage when switched to earth. This introduces severe constraints on the design of the electronic switches

Note that these could be corrected in WRN by moving the switches - however, the particular circuit shown is bad.

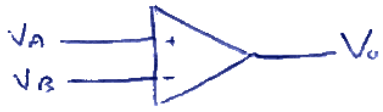
14.3 A/D CONVERTORS

We shall examine three basic types:

- staircase ramp A/D
- successive approximation A/D
- dual slope integration A/D.

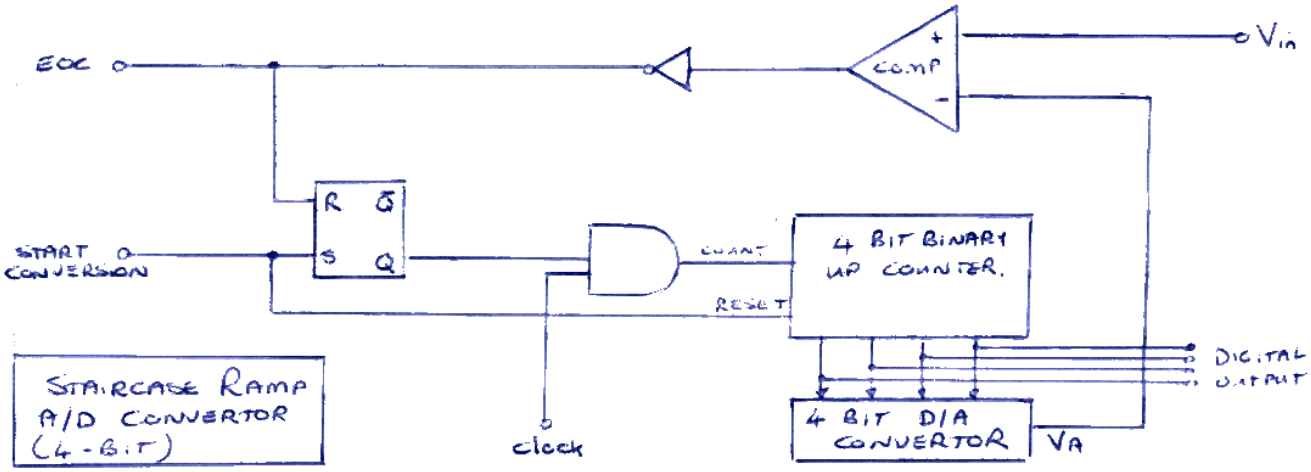
We shall be using comparators; their operation is as follows

If $V_A > V_B$ then V_O is high
If $V_A < V_B$ then V_O is low



In fact the comparator is an operational amplifier with no feedback whose output voltage swing is limited to the two logic extremes.

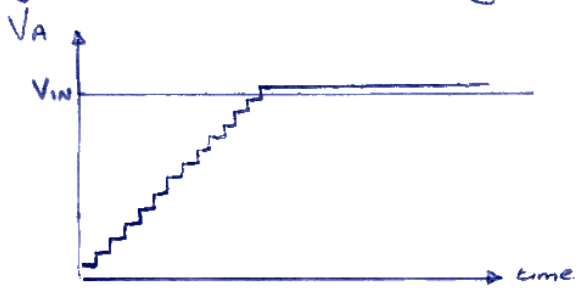
14 3.1 STAIRCASE RAMP A/D.



Principle of operation counter output is converted to an analogue voltage and compared to the input voltage. The counter is incremented by a clock signal until the analogue equivalent of its output exceeds the input voltage.

Operation • when 'START CONVERSION' goes high it clears the counter which will be unable to count until it goes low again. Q is also set so the clock pulses can get through to the counter. The D/A output is zero < V_{IN} so the comparator is high and EOC and R are low.

• when 'START CONVERSION' goes low, the counter begins counting. When $V_A > V_{IN}$ the comparator goes low, R goes high setting Q low and disabling the clock input to the counter.



OUTPUT OF D/A IN STAIRCASE RAMP A/D

Disadvantages:

The conversion time is dependent on the input voltage - we must allow for the worst case. The accuracy & linearity are limited by the D/A convertor. SR A/D is also sensitive to changes in the input signal during conversion. The effect of this is to

introduce uncertainty in the time of conversion. If this is a problem it can be overcome by putting the analogue input into a sample-and-hold amplifier synchronised to the start conversion pulse.

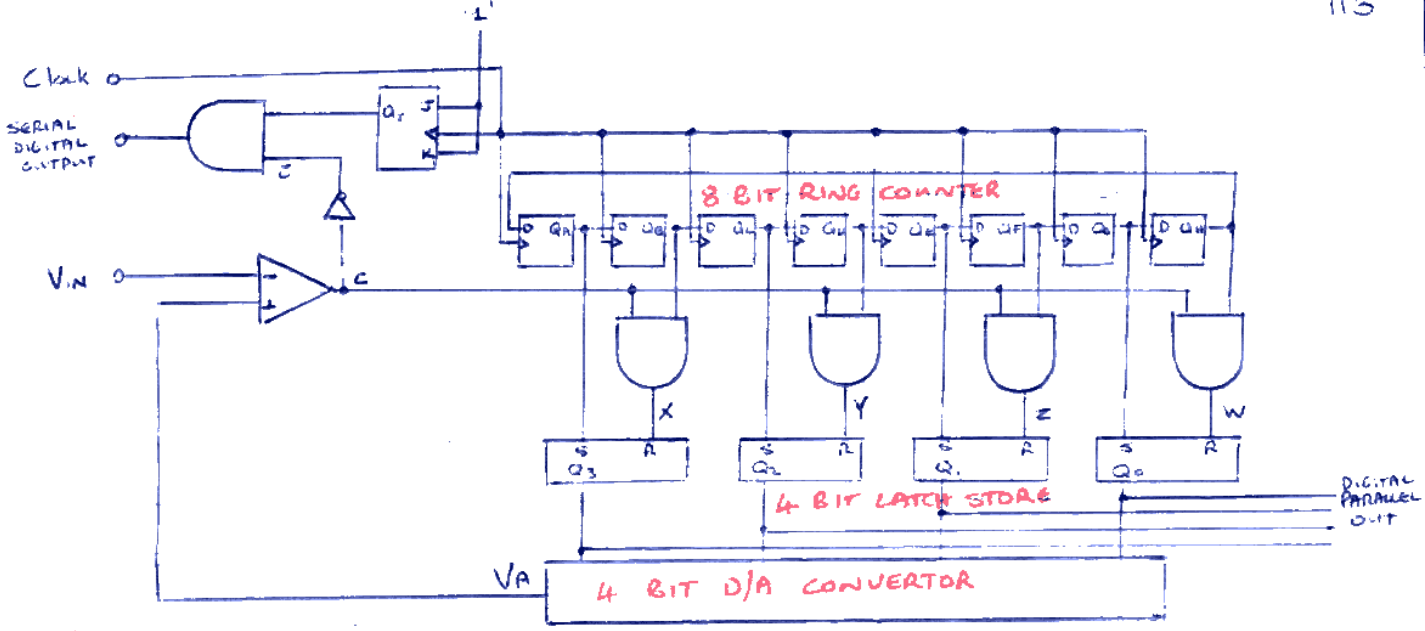
SR is very sensitive to noise as it converts a particular signal at one instant in time. To overcome this the input signal must be filtered by analogue circuitry before being input to the convertor.

14.3.2 SUCCESSIVE APPROXIMATION A/D.

Conversion time is independent of input, furthermore a serial output is obtainable with minimal additional circuitry.

Principle of operation: input voltage compared with a successively narrowed range of analogue voltages until it can be located within one step size where the input voltage lies. Each of these ranges of analogue voltages is identified by one bit in the digital output. The voltages for comparison are generated by a D/A convertor, the input of which is successively set by incrementing the next bit in the digital number.

Operation: The initialisation circuitry (not shown) sets the 8-bit ring counter to $\overset{Q_A}{1}0000000^{Q_H}$, and sets the four bit latch to $\overset{Q_3}{0}000^{Q_0}$. Notice however that the S input of Q_3 is set by Q_A of the shift register so the latch is set to 1000. The first comparison is to compare V_{in} with $V_F/2$ where V_F is the full scale voltage of the D/A converter ($V_F/2$ corresponds to 1000). If V_{in} is greater than $V_F/2$ the comparator output is logic 0 else 1. We consider the two cases:



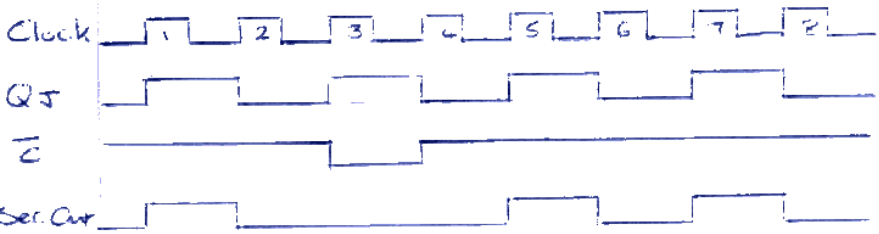
SUCCESSIVE APPROXIMATION A/D

- if $V_{IN} > V_F/2$, then $C = 0$, the and gates are disabled. When the clock pulse causes 1 to shift from Q_A to Q_B, R of Q_3 remains zero so Q_3 remains set.
- if $V_{IN} < V_F/2$, then $C = 1$, the AND gates are enabled. When clock pulse causes shift, gate X goes high clearing Q_3 to 0. (and VA goes to zero).

At the next clock edge, the 1 is shifted from Q_B to Q_C so Q_2 is set. Thus VA is $V_F/4$ or $3V_F/4$, depending on Q_3, and this is used as the comparison voltage, etc. until all 4 bits have been successively set or cleared. Each comparison takes two clock pulses ⇒ conversion time is 8 clock pulses.

In general for an N-bit convertor the conversion takes 2N pulses (staircase ramp was 2^N max)

To see how the serial digital output is done, consider the timing diagrams for a specific analogue input. We shall assume that the analogue input gives rise to a digital output from the convertor of 1011. The timing diagrams overleaf are applicable: note that Q_J is used to divide the clock pulse by 2 to ensure a correct conversion rate.



TIMING DIAGRAM FOR SUCCESSIVE APPROXIMATION A/D

The leading edge of each clock pulse cause the 1 in the ring counter to be shifted to the next flip flop.

Pulse 1: Q_3 set $\Rightarrow V_A = V_F/2 < V_{IN} \Rightarrow C = 0 \Rightarrow \bar{C} = 1 \Rightarrow$ ser. out = 1

Pulse 2 C low $\Rightarrow Q_3$ remains set $\Rightarrow$ C remains low

But Q_J has toggled so ser. out goes low to give an inter bit space

Pulse 3: Q_2 set $\Rightarrow V_A = 3V_F/4 > V_{IN} \Rightarrow \bar{C}$ low $\Rightarrow$ ser out = 0.

Pulse 4 Q_2 reset as C high $\Rightarrow$ C low, Q_J low $\Rightarrow$ inter-bit pause

Pulse 5. Q_1 set, C low $\Rightarrow$ ser. out. high ... etc

Thus if a receiving system strobed the serial output at the falling edge of every odd numbered pulse it would get the A/D output serially in the order MSB to LSB.

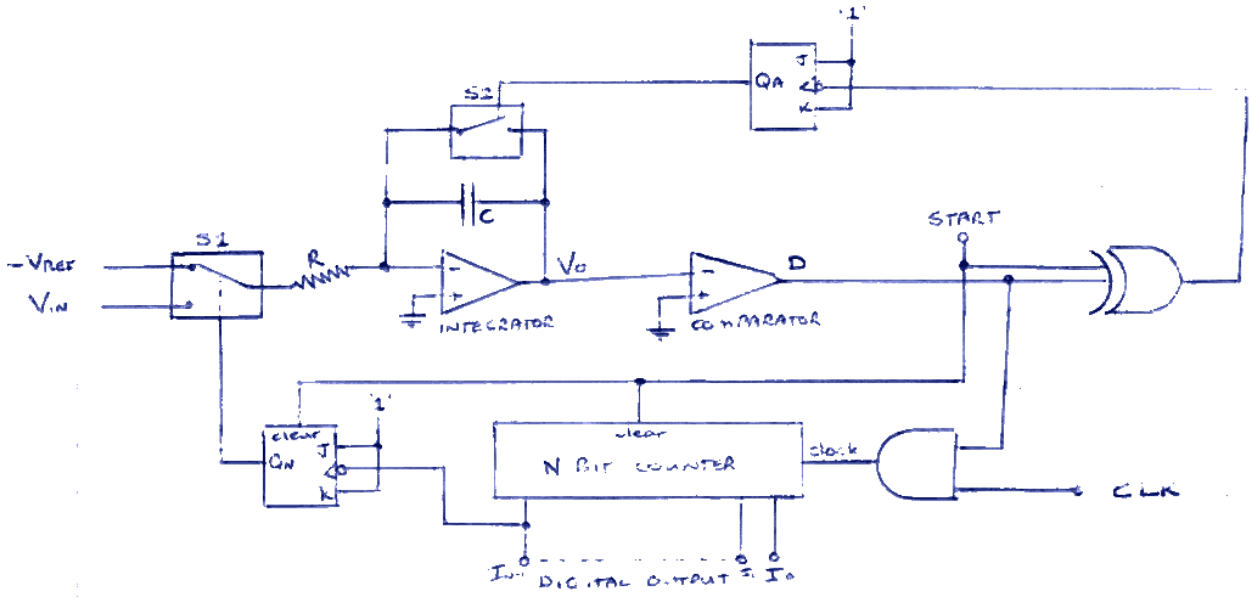
The SA suffers from the same limitations (except conv. time) as SR converters, namely:

- limitations on accuracy and linearity caused by D/A
- noise sensitivity
- uncertainty during conversion.

However, SA is cost effective and high-speed, and thus popular

14.3.3 DUAL SLOPE INTEGRATION A/D

The DSI overcomes the accuracy limitations of the previous two at the expense of conversion time. S_1 & S_2 are digitally controlled electronic switches; S_1 is controlled by Q_N, S_2 by Q_A. If $Q_N = 1$ then S_1 is switched to voltage $-V_{REF}$, otherwise to V_{IN}. If $Q_A = 1$ then S_2 is open circuit, otherwise it is closed circuit. Q_A, Q_N are -ve edge-triggered F-F's



DUAL SLOPE INTEGRATION A/D

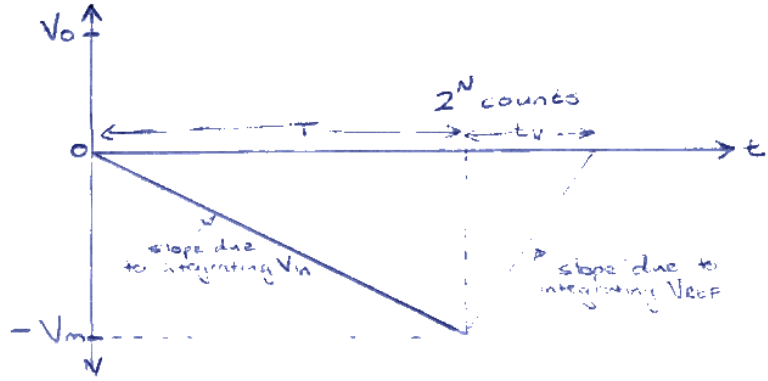
The output voltage Vo of the integrator is determined by the integral of the input voltage, providing S2 is open (ie QA = 1). If S2 is close then Vo is zero due to virtual earth at op-amp input. Prior to the start of the conversion, the 'start' line is low, as is QA so Vo is zero and hence D as well. Thus the AND gate is disabled, no clock pulses reach the counter - the circuit is stable.

Principle of operation: integrate input voltage for a fixed period of time (given by a full scale count of the counter) and then to integrate a known reference voltage of opposite polarity until the integrator output returns to zero. The time to do this is measured by the counter and is equivalent to the average input voltage over the initial fixed period of integration (ie the converter only converts +ve voltages)

To initiate the conversion the 'start' line goes high, clearing the counter and QN, switching S1 to VIN. Despite X also going high QA stays constant as it is -ve edge triggered. Thus S2 remains closed, D remains low. The circuit is stable and ready to start conversion - to actually initiate conversion start must go low again.

On the falling edge of the start pulse, X goes low toggling Q_A to '1' opening S_2. The integrator begins integrating $-V_{IN}$, resulting in -ve V_o, high D, AND gate enabled and clock pulses reaching counter which begins counting. This situation does not change until 2^N clock pulses have occured, at which stage the counter is reset to zero. This is the only time in counting when J_{N-1} has a falling edge; this toggles Q_N to '1' switching S_1 to $-V_{REF}$. At this point the op amp output voltage has reached a negative voltage $-V_m$. The output begins rising, but is still -ve thus D remains high and the counter continues counting. Providing the input voltage is within range (ie $V_{IN} < V_{REF}$), then within 2^N counts, the op amp output V_o will return to zero causing D to go low. This disables the AND gate freezing the counter (at the digital equivalent of V_{IN}), and causes X to go low, toggling Q_A low which closes S_2 and clamps V_o back to earth. The circuit is thus in a fixed stable condition until the next clock pulse arrives

To derive the relationship between the output and input of the convertor, we consider V_o as a function of time



INTEGRATOR OUTPUT VOLTAGE FOR DSI

We assume that V_{IN} is constant (although this need not be the case). The integration of V_{IN} gives

$$(-V_m) - 0 = \frac{1}{RC} \int_0^T (-V_{IN}) . dt$$

This gives $V_m = \frac{T}{RC}\left[\frac{1}{T}\int_0^T V_{IN}\,dt\right]$. The expression in brackets corresponds to $\overline{V_{IN}}$ (the mean value of V_{IN} over the integration period)

So $V_m = \overline{V_{IN}}\ 2^N \tau/(RC)$ ($T = 2^N\tau$, τ the clock period)

When $-V_{REF}$ is integrated, the integrator output voltage V_o goes from $-V_m$ to 0, giving

$$0 - (-V_m) = \frac{1}{RC}\int_0^{t_v} -(-V_{REF})\,dt$$

which gives

$$V_m = \frac{t_v}{RC}\cdot V_{REF} \quad \text{since } V_{REF} \text{ is constant}$$

Now $t_v = B\tau$ where B is the converter output (digital).

Thus

$$V_{REF}\cdot B\tau/(RC) = \overline{V_{IN}}\ 2^N\tau/(RC)$$

so

$$B = \overline{V_{IN}}\ 2^N/V_{REF} = \overline{V_{IN}}/\delta V \quad \text{where } \delta V = V_{REF}/2^N$$

Note

- the output is directly proportional to the average input
- the absolute value of the digital output is dependent only on the V_{REF} voltage
- since we integrate the input voltage we can get excellent noise rejection (eg consider a 50Hz (mains) noise source; by letting T be a multiple of 20ms we get complete noise rejection of 50Hz noise and its harmonics)
- conversion time is $2^N + B$ clock periods. Worst case is 2^{N+1}. Thus the maximum sampling rate is half that of the staircase ramp converter.

14.4 D/A AND A/D SPECIFICATIONS.

14.4.1 D/A SPECIFICATIONS

14.4.1.1 RESOLUTION (STEP-SIZE)

The resolution δV is the smallest change in analogue output given by $\delta V = V_F / 2^N$ where V_F is the full-scale output (never actually produced - outputs range from 0 to $(2^{N-1})\delta V$)

14.4.1.2 ACCURACY

Ideally $V = D \cdot \delta V$. We measure the maximum deviation of the output for each input from the straight line given by $V = D \cdot \delta V$. If this is ε, we define the converter (in)accuracy as $\pm \varepsilon$. Usually this is expressed as a fraction of step size and given in units of LSBs. The accuracy depends on the zero (determined by trimming op amp) and fullscale (determined by accuracy of V_{ref}) errors as well as linearity.

14.4.1.3 LINEARITY

A converter is linear if equal changes in input produce equal changes in output. We measure the non-linearity as the deviation of the converter from the straight line drawn between the first and last measureable points of the converter (sometimes between second and last points in case). Nonlinearity is a result of inaccuracies in voltage drops across eg switches/resistors, and is also influenced by temperature and time drift.

4.4.1.4 SETTLING TIME - the time taken for the output to reach some percentage of its final value after a change in input occurs.

4.4.2 A/D SPECIFICATIONS

4.4.2.1 QUANTIZATION ERROR - the error resulting from quantising the input, $\pm\frac{1}{2}\delta V$ where δV is the step size.

4.4.2.2 ACCURACY - as for D/A.

4.4.2.3 LINEARITY - as for A/D.

4.4.2.4 SAMPLING RATE - maximum rate at which data can be sampled. For signal clock rate and N bits, this is approx

$2N$ for successive approximation

2^N for staircase ramp

2^{N+1} for dual slope integration

4.4.2.5 NOISE REJECTION

4.4.2.5 Noise rejection: - This is the ability of the device to ignore input noise signals (i.e. to attenuate these noise signals). There are 2 types of noise signal (i) Normal mode (which is in series with the signal) and (ii) common mode (in parallel with the signal). The normal mode noise signal is shown as V_N in figure 4.15 and would be added to the analogue input voltage we are converting. The common mode noise signal is shown as V_C in figure 4.15 and it is added to both signal leads. If we have floating input leads then the differential common mode noise signal to the convertor input is zero and thus complete common mode noise rejection is achieved. If however the one input lead is earthed at the convertor then the common mode noise signal, V_C is not added to this lead but it is added to the other lead and thus the full common mode noise signal is added to the input voltage.

Normal mode noise rejection can be achieved by filtering the input signal. This is done automatically in the Dual Slope Intergration A/D convertor by integrating the input voltage over a fixed period of time. If we choose this fixed period to be a multiple of 20 ms then total rejection of normal mode mains noise and its harmonics is achieved (for a 50 hz mains).

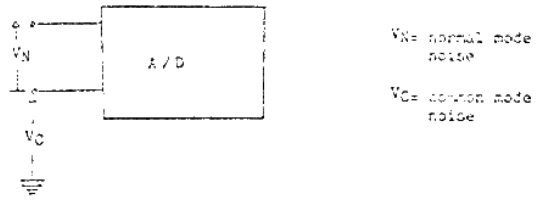


Figure 4.15 Common and Normal Mode noise signals