

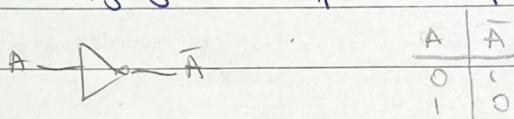
Logic gates and inverters

Basic properties of logic gates

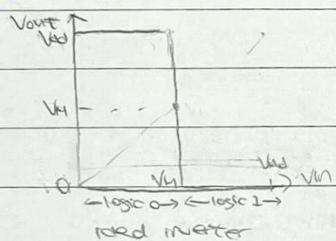
- A logic gate should have the following properties:
 - ↳ The binary op gives the req. logic function of the i/p(s)
 - ↳ Quantisation of amplitudes is req. \rightarrow nonlinear devices
 - ↳ Signal amplitudes should be regenerated in passing through a gate \rightarrow significant gain
 - ↳ Capable of accepting more than one i/p (no. of independent i/p's is the fan-in)
 - ↳ Capable of driving more than one o/p (no. of independent o/p's driven is the fan-out)
 - ↳ Changes at the o/p of the gate do not affect the signals at the i/p

Operation and basic properties of inverter

- An inverter is the simplest logic gate we can use and one of



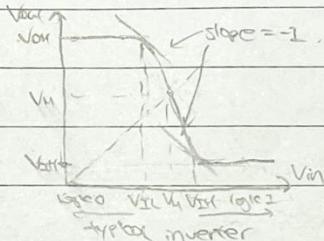
- The transfer characteristics of an ideal and typical inverter is as follows



$\hookrightarrow V_{in} = 0, V_{out} = V_{dd}$; $V_{in} = V_{dd}, V_{out} = 0$

↳ infinitely sharp transition at $V_{in} = V_m$

↳ zero propagation delay



→ I/P and O/P voltages deviate from ideal values

↳ gradual transitions at $V_{in} = V_{th}$.

↳ finite propagation delay.

- V_{IH} : min. voltage recognised as logic 1 ; V_{IL} : max. voltage recognised as logic 0.

V_{out}: min. o/p voltage when o/p is HIGH

; Vol: max. dp voltage when o/p is low.

- The inverter switching threshold V_H is defined as the pt. at which the voltage transfer curve intersects w/ the line $V_{out} = V_{in}$ \rightarrow represents the pt. at which the inverter switches state. (usually $V_H = \frac{V_{in}}{2}$, but not always)

Designing "real" logic gates

- The V_H and V_L signals must be consistent
 - ↳ If two gates of the same logic family are connected, V_H > V_H_T for HIGH to be recognised as logic 1 and V_L < V_L_T for LOW to be recognised as logic 0
 - V_L, V_H, V_L_T, V_H_T characterise the behaviour of a logic family
 - We only req. V_H ≤ V_H_T - ε or V_L ≥ V_L_T + ε to be recognised if it were off for the presence of noise.

For Personal Use Only -bkwk2

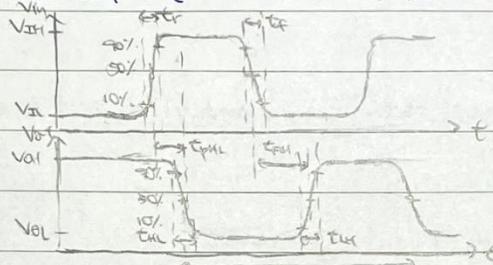
Noise and noise margins

- Noise is spurious signals generated in a system mainly due to inductive or capacitive coupling between signal lines.
- If the noise amplitude is large enough, it may change the signal enough to change its logic level.
- Noise margin is a measure of how much noise can be added to the logic levels of a circuit w/o the circuit responding improperly, and depend on the transfer characteristics
 - ↳ Low state noise margin : $NM_L = V_{IL} - V_{OL}$
 - ↳ High state noise margin : $NM_H = V_{OH} - V_{IH}$
- For the gates to work correctly and w/o spurious d/s, we req. $NM_L > 0$ and $NM_H > 0$.
- The logic swing is the expected range of signals at the o/p., $\text{logic swing} = V_{OH} - V_{OL}$



Rise time, fall time and delay time

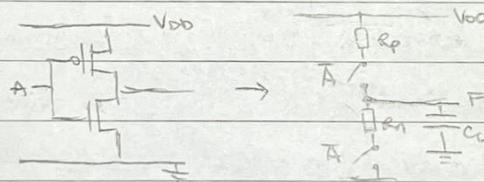
- The rise time, fall time and delay time are defined as follows:



t_r : rise time (μs) ; t_f : fall time (μs) ; t_{PHL} : fall time (high-low) ; t_{PLH} : rise time (low-high) ; t_{cyc} : cycle time.

t_{PHL} : propagation time high-low ; t_{PLH} : propagation time low-high ; t_{cyc} : cycle time.

- Propagation delays of CMOS gates can be determined by analysing RC equivalent circuits

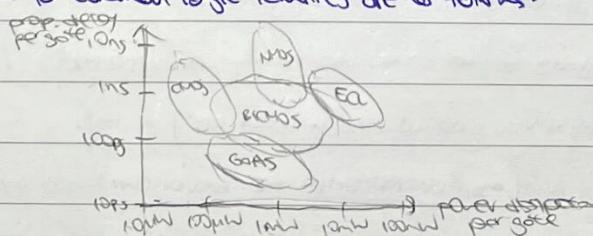


For a pure RC circuit, $\text{rise time} = 2.2RC$, $\text{fall time} = 2.2RC$.

Power-delay product

- The power-delay product is the product of power consumption (averaged over a switching event) times the i/p-o/p delay.

- The power-delay product for common logic families are as follows:

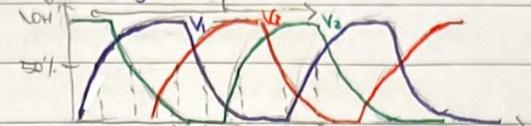
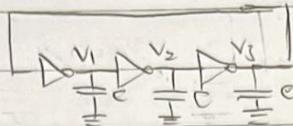


For Personal Use Only -bkwk2

Application of inverters

① Ring oscillator

- A ring oscillator is formed by connecting an odd no. n of inverters connected in chain.
- The ring oscillator is a fundamental circuit for evaluating the intrinsic speed of a CMOS logic process.



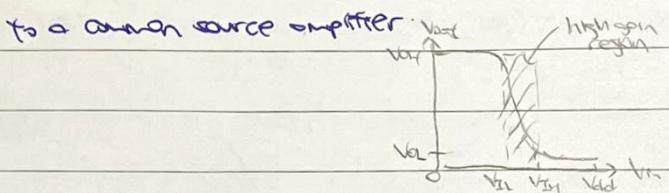
Assuming the time delay for each inverter is const, i.e., $t_{PHL} = t_{FALL} = t_{PAH} = t_{PHH}$, $T = t_{PHL} + t_{FALL} = t_{PHH}$

$$f = \frac{1}{T} = \frac{1}{n(t_{PHL} + t_{FALL})}$$

↳ We can also define the propagation delay t_p to be $t_p = \frac{1}{2}(t_{PHL} + t_{FALL})$

② Amplifier

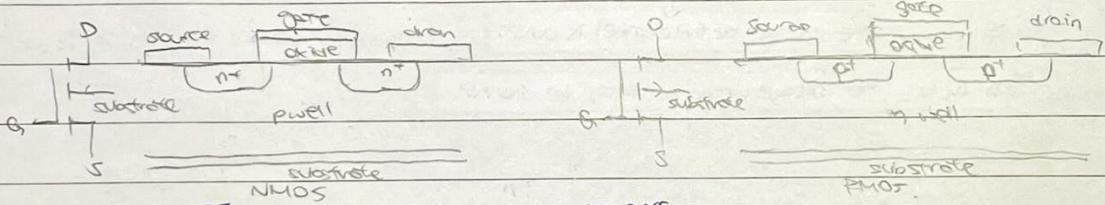
- For V_{in} between V_{IL} and V_{IH} , the inverter gain > 1 . → Inverter can act as linear amplifier.
- Applying a resistive load to the inverter and operating the inverter in this region is equivalent



MOS transistor and logic circuits

MOS transistor

- The structure of a MOS transistor is as follows:



- The NMOS/PMOS contains the following key elements:

- ↳ Si substrate containing p-type/n-type impurities
- ↳ Two heavily doped n⁺/p⁺ regions (source, drain regions).
- ↳ Inversion layer under gate to connect source and drain. (depends on gate voltage)
- ↳ Gate oxide is a thin film of insulating SiO₂.

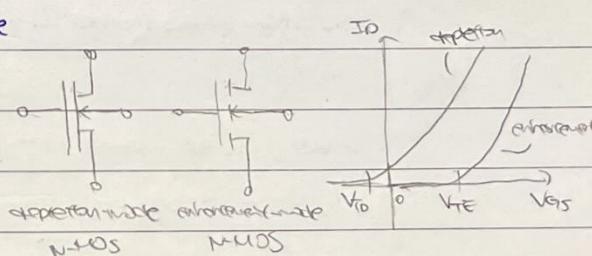
- The flow of current between source and drain is controlled by the gate voltage.

(Gate voltage modulates the conductivity of the inversion layer).

- MOSFETs can be depletion mode or enhancement mode

↳ Depletion mode: $V_{t,D} < 0 \rightarrow$ normally ON

↳ Enhancement mode: $V_{t,E} > 0 \rightarrow$ normally OFF



For Personal Use Only -bkwk2

MOS transistor operation (assume $V_S = V_D = 0$)

(1) Accumulation ($V_{GS} < 0$, only V_{GS})

- Source to drain path consists of two reverse-biased diodes \rightarrow no current

(2) Depletion / cut-off ($V_{GS} < V_T$, only V_{GS})

- Electrostatic field attracts e^- towards the gate, but there is still a deficit of e^- (so channel is highly resistive) \rightarrow no drain current con flow.

(3) Inversion, linear/triode ($V_{GS} > V_T$, $0 \leq V_{DS} \leq V_{D,sat} = V_{GS} - V_T$)

- An inversion layer is induced in the channel and connects source and drain.

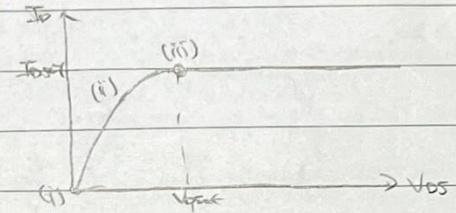
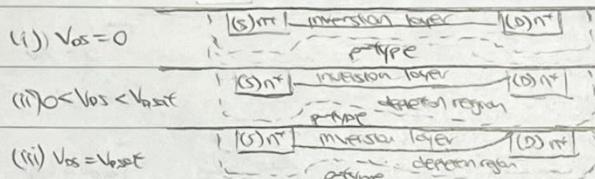
The induced channel acts like a resistor \rightarrow current depends linearly on drain voltage V_D .

- As V_{DS} increases, the depletion region widens and e^- conc. in the inversion layer decreases

on the drain side \rightarrow channel conductance decreases $\rightarrow I_D - V_D$ slope decreases.

- The inversion layer eventually vanishes on the drain side \rightarrow pinch off, $I_D - V_D$ curve is flat.

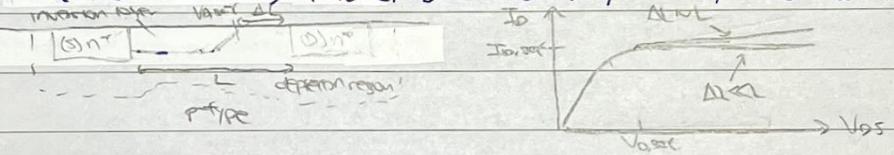
(Saturation). V_{DS} has reached the saturation voltage $V_{D,sat}$ and I_D is the saturation current $I_{D,sat}$.



(4) Inversion, saturation ($V_{GS} > V_T$, $V_{DS} \geq V_{D,sat} = V_{GS} - V_T$)

- As V_{DS} increases above $V_{D,sat}$, the channel length L effectively changes by ΔL . The region of the channel ΔL is depleted \rightarrow highly resistive \rightarrow voltage increase in $V_{DS} > V_{D,sat}$ are dropped across this part.

- If $\Delta L \ll L$ we have const. $I_D = I_{D,sat}$; If $\Delta L \sim L$, I_D increases slightly w/ V_{DS} above $I_{D,sat}$.



$I_D - V_D$ characteristics.

- For n-type MOSFET, when there is a continuous channel, i.e. $V_{GS} > V_T$, $0 \leq V_{DS} \leq V_{GS} - V_T$ [linear]

$$I_D = \frac{k}{2} (2(V_{GS} - V_T)V_{DS} - V_{DS}^2) \quad \text{for very small } V_{DS}, \text{ we approx. } I_D = k(V_{GS} - V_T)V_{DS} \rightarrow R_D = \frac{1}{k(V_{GS} - V_T)}$$

For n-type MOSFET, when the channel is pinched-off, i.e. $V_{GS} > V_T$, $V_{DS} \geq V_{GS} - V_T$ [saturation].

$$I_D = \frac{k}{2} (V_{GS} - V_T)^2$$

- Ideally, I_D is independent of V_{DS} in the saturation region, but increasing V_{DS} causes an electronic shortening of the channel \rightarrow increase in drain current (Early effect).

$$I_D = \frac{k}{2} (V_{GS} - V_T)^2 (1 + \lambda V_{DS})$$

where λ is the drain length modulation coefficient, ($V_A = \frac{1}{\lambda}$ is the Early voltage).

- k is the transconductance parameter

For Personal Use Only -bkwk2

Transconductance parameter K .

- The transconductance parameter K depends on factors related to physical and electrical characteristics of the device material, and dimensions of the conductive channel, (length L , width W).

$$K = K\left(\frac{W}{L}\right) = \mu C_{ox} \left(\frac{W}{L}\right) = \mu \frac{\epsilon_{SiO_2}}{t_{ox}} \left(\frac{W}{L}\right)$$

where ϵ_{SiO_2} is the permittivity of the SiO_2 gate insulator,

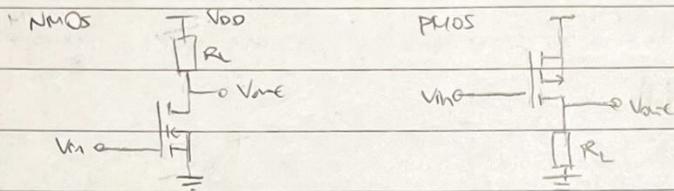
t_{ox} is the thickness of the SiO_2 gate insulator.

] determined from material characteristics.

M is the mobility of the charge carrier ← obtained empirically from electrical measurement

NMOS and PMOS inverter

- The circuit diagram for the NMOS inverter and the PMOS inverter is as follows:



The resistor R_L forms a load. The pd. across R_L is determined by the current driven through it by the MOSFET (the driver).

- Desirable values of R_L would be

↳ A high value to reduce power consumption and bring V_{out} towards $0/V_{DD}$ for NMOS/PMOS inverter.

↳ A low value to ensure V_{out} approaches $V_{DD}/0$ (NMOS/PMOS) rapidly when the o/p terminal is loaded w/ a capacitance. ↳ reduce t_{PLH}/t_{PHL} (NMOS/PMOS).

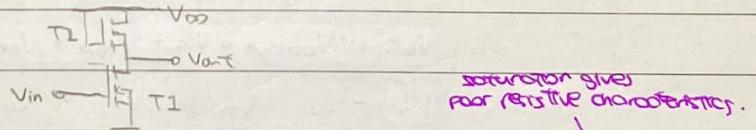
→ req. are in conflict so only choice is a compromise.

- Voltage transfer characteristic can be gained by plotting I_D vs V_{DS} for the resistor and the transistor then finding intersections for various V_{GS} . Alternatively, directly equate the load and driver currents.

general strategy - assume the regime the transistor is operating if unknown and check if the answer is contradictory.

NMOS inverter w/ NMOS load

- Load resistors are much larger than transistors → not economical to fabricate → use transistor as load.



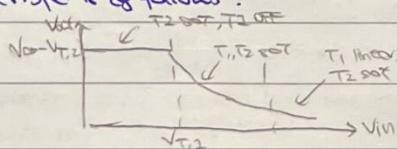
Transistor behaves like a pseudo-resistor w/ its gate connected to the drain (always in saturation)

- A much smaller $\frac{W}{L}$ ratio (smaller k) for the load device to enhance its resistive properties → reduced logic.

o/p in HIGH state → load must conduct, i.e. $V_{GS,2} \geq V_{T2}$; $V_{out} = V_{DD} - V_{DS,2} = V_{DD} - V_{GS,2} - V_{T2} \geq V_{DD} - V_{T2} \rightarrow V_{out}$ reduced.

o/p in LOW state → $V_{GS,2}$ is max → resistance of load min. → opposite of what we want (poor performance)

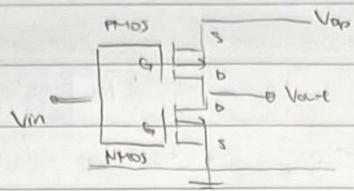
- The typical transfer characteristic is as follows:



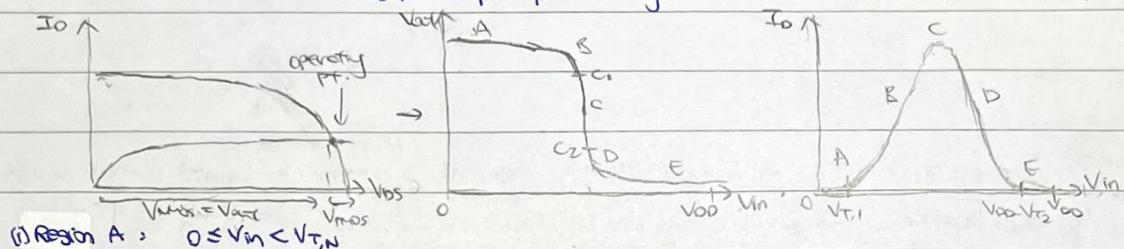
For Personal Use Only -bkwk2

CMOS inverter

- The circuit diagram for the CMOS device is as follows



- The output is pulled to one of the power rails by a conductive (on) device. The other device serves as a load of effectively infinite resistance \rightarrow static properties closer to ideal inverter.
- The challenge is that PMOS devices are rel. slow ($\mu_p < \mu_n$) \rightarrow device needs to be physically larger to compensate, \rightarrow more complex fabrication process.
- The voltage transfer characteristic can be found by plotting the NMOS and flipped PMOS characteristics on the same V_{in} vs V_{out} plot, and finding the intersection pts (for various V_{in} s)



(i) Region A: $0 \leq V_{in} < V_{T,N}$

\hookrightarrow NMOS OFF, PMOS ON (linear)

\hookrightarrow NO current flows (since NMOS is OFF).

(ii) Region B: $V_{T,N} \leq V_{in} < V_M$

\hookrightarrow NMOS ON (saturated), PMOS ON (linear)

\hookrightarrow NMOS starts to conduct

(iii) Region C: $V_{in} = V_M = V_{out}$

\hookrightarrow NMOS ON (saturated), PMOS ON (saturated)

$$I_{DN} = I_{DP} \rightarrow k_N (V_{in} - V_{TN})^2 = k_P ((V_{DD} - V_{in}) - |V_{TP}|)^2 \rightarrow V_M = \frac{V_{DD} - |V_{TP}| + V_{TN} \sqrt{k_N / k_P}}{1 + \sqrt{k_N / k_P}}$$

\hookrightarrow If $k_P = k_N$ and $V_{TN} = |V_{TP}|$, then $V_M = \frac{V_{DD}}{2}$

\hookrightarrow Values of pts C₁ and C₂ can be found using the V_{in} above and

$$C_1: |V_{DS1}| = |V_{GS1}| - |V_{TP}| = V_{DD} - V_{in} - |V_{TP}| \rightarrow V_{out} = V_{DD} - |V_{DS1}| = V_{in} + |V_{TP}|$$

$$C_2: |V_{DS2}| = |V_{GS2}| - V_{in} = V_{in} - V_{TN} \rightarrow V_{out} = |V_{DS2}| = V_{in} - V_{TN}$$

(iv) Region D: $V_M < V_{in} \leq V_{DD} - |V_{TP}|$

\hookrightarrow NMOS ON (linear), PMOS ON (saturated)

(v) Region E: $V_{DD} - |V_{TP}| \leq V_{in} \leq V_{DD}$

\hookrightarrow NMOS ON (linear), PMOS OFF

\hookrightarrow No current flows (since PMOS is OFF)

* V_{TH} and V_{IL} can be found from $\frac{dV_{out}}{dV_{in}} = 1 \rightarrow$ we find $NM_1 = NM_2 \approx 0.5V_{DD}$.

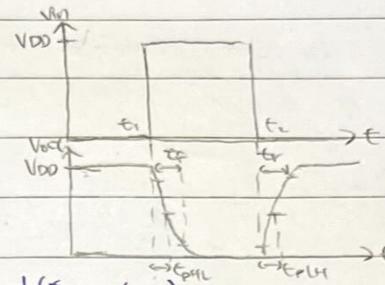
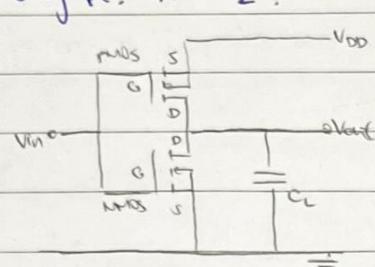
* Alternatively, the voltage transfer characteristic / current draw can be found by projecting the line of intersection of current surfaces $I_{DN} = f_1(V_{in}, V_{out})$, $I_{DP} = f_2(V_{in}, V_{out})$

For Personal Use Only -bkwk2

CMOS inverter switching speed

- Assume the input signal is ideal — MOSFETs snap in and out conduction instantaneously.

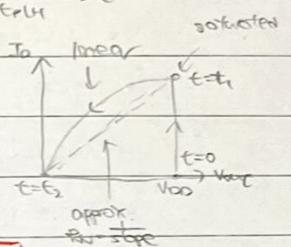
- Propagation delay arises from the time taken for load capacitance C_L to be charged/discharged to the switching pt. $V_M = \frac{V_{DD}}{2}$.



- We define the average propagation delay t_p as $t_p = \frac{1}{2}(t_{PHL} + t_{PLH})$

- Consider the average channel resistance R_N

$$R_N = \frac{V_{DD}}{I_D N (V_{DD} - V_{TN})^2}$$



It can be shown that

$$t_{PHL} \propto 0.7 R_N C_L \approx \frac{C_L}{k_N V_{DD}}$$

$$t_{PLH} \propto 0.7 R_P C_L \approx \frac{C_L}{k_P V_{DD}}$$

thus the average propagation delay is $t_p = \frac{1}{2}(t_{PHL} + t_{PLH}) = \frac{C_L}{2V_{DD}} \left(\frac{1}{k_N} + \frac{1}{k_P} \right)$

- Similar calculations yield the rise time t_r and fall time t_f

$$t_r \approx 0.5 \frac{C_L}{k_N}$$

$$t_f \approx 0.5 \frac{C_L}{k_P}$$

where $V_{DD} \approx 5V$, $V_{TN} = |V_{TH}| = 1V$.

* Capacitor discharge through NMOS (so t_{PLH}, t_f depend on k_N)

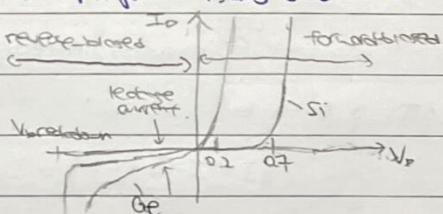
Capacitor charge through PMOS (so t_{PHL}, t_r depend on k_P)

* For gates other than NOT, we may have two NMOS/PMOS \rightarrow series: $(\frac{W}{L})_{eff} = \frac{W}{2L}$, parallel: $(\frac{W}{L})_{eff} = \frac{2W}{L}$

Bipolar transistors and logic circuits

Diodes.

- The diode is formed by a doped PN junction. Its on-state characteristics are



The current of an ideal diode is given by

$$I_D = I_S [e^{\frac{V_D}{V_T}} - 1] \approx I_S e^{\frac{V_D}{V_T}}$$

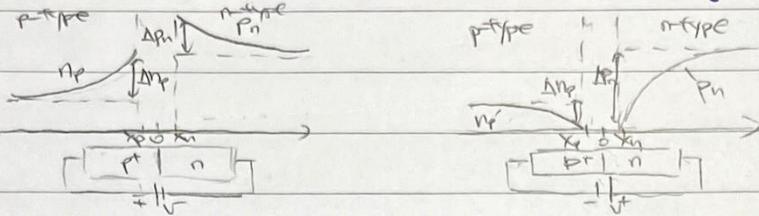
where $V_T = \frac{kT}{q}$ is the thermal voltage ($V_T = 25mV$ at RT)

and I_S is the reverse saturation current. ($I_S \sim 10^{-5} - 10^{-4} A$ for IC diodes)

For Personal Use Only -bkwk2

Transient behaviour of diodes

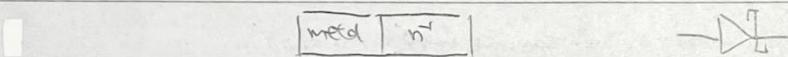
- When the diode is forward-biased, minority carriers are injected from other side of the junction. Current is carried by majority carriers.
- When the diode is reverse-biased, only minority carriers contribute to current flow (leakage current). Minority carrier current is increased where additional h-e pairs are generated (e.g. heat, light).



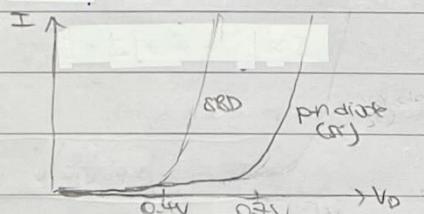
- The time to switch between conductivity and nonconductivity is determined by the time taken to establish minority carrier distributions.
 - ↳ Turn ON : diode switch into conduction rel. quickly — time taken determined by circuit capacitance
 - ↳ Turn OFF : switch-off is slow as injected minority carriers need to be removed — this can be represented by a diffusion capacitance $C_d = A I_0 \cdot \frac{2}{V_c} t$ | t is mean recombination time of carriers,

Schottky barrier diode (SBD)

- The SBD is a device w/ the properties of a diode created by a microscopically clean contact between a lightly-doped semiconductor and certain metals \rightarrow unipolar device.



- A characteristic potential barrier ϕ_B at metal-semiconductor interface impedes flow of e^- from metal to semiconductor. Height of barrier depends on the materials used (typically 0.6-0.8V)
- A depletion region is set up in the semiconductor close to the surface.
- e^- in the semiconductor may acquire energy sufficient to surmount the barrier and escape to the metal \rightarrow small leakage current.
- Forward bias : height of barrier reduced $\rightarrow e^-$ flow more easily from metal to semiconductor
- Reverse bias : height of barrier increased and depletion region wider.
- The I-V characteristics of the SBD is as follows



The current for SBD is similar to that of a pn junction diode

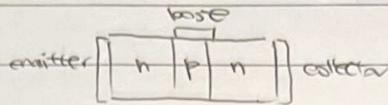
$$I = I_0 [e^{\frac{qV_d}{kT}} - 1] \approx I_0 e^{\frac{qV_d}{kT}}$$

A SBD of a given size typically has a lower forward voltage than a comparable pn junction diode operating at the same forward current (threshold voltage V_T no.3V smaller)

For Personal Use Only -bkwk2

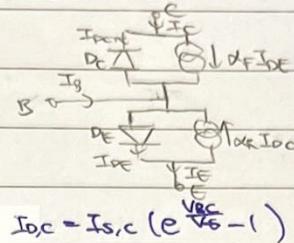
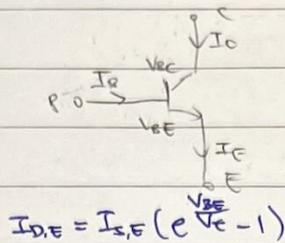
Bipolar transistors

- The bipolar transistor is formed by two back-to-back pn junctions.



- The operation of the BJT can be described mathematically using Ebers-Moll model.

(Ebers-Moll model applies to all modes of operation of the device)



$$I_E = I_{D,E} - \alpha_F I_{D,C}$$

$$I_C = \alpha_F I_{D,E} - I_{D,C}$$

where $I_{S,E}$, $I_{S,C}$ are the saturation currents of the collector-base, emitter-base junctions, respectively

and $\alpha_F \approx 1$, $\alpha_R < 0.5$.

$$\Rightarrow I_E = I_{S,E} (e^{\frac{V_{BE}}{V_T}} - 1) - \alpha_R I_{S,C} (e^{\frac{V_{CE}}{V_T}} - 1) ; I_C = \alpha_F I_{S,E} (e^{\frac{V_{BE}}{V_T}} - 1) - I_{S,C} (e^{\frac{V_{CE}}{V_T}} - 1)$$

- For an ideal transistor, reciprocity theorem gives $\alpha_F I_{S,E} = \alpha_R I_{S,C} = \alpha_S$

Bipolar transistor operation.

① Cut-off ($V_B < V_C, V_E > V_F$)

↳ behaves like an open switch ($I_B = I_C = 0$)

↳ B-C junction, B-E junction reverse biased.

② Forward active ($V_B < V_C, V_E > V_F$)

↳ behaves like an amplifier

↳ B-C junction reverse biased, B-E junction forward biased

$$I_C = \beta_F I_B$$

③ Saturation ($V_B > V_C, V_E > V_F$)

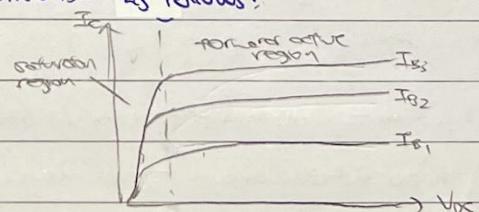
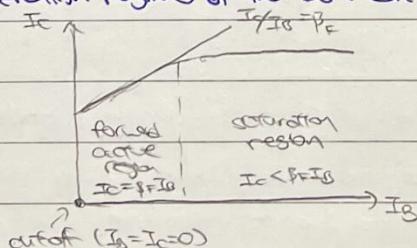
↳ behaves like a closed switch

↳ B-C junction, B-E junction forward biased

$$I_C < \beta_F I_B , \text{ degree of saturation } \sigma = \frac{I_C}{\beta_F I_B} < 1$$

* There is also the reverse-active region, but this operation regime is not very useful.

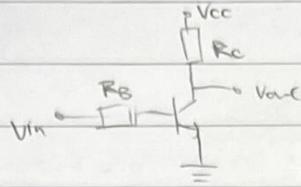
- The operation regimes of the BJT can be visualised as follows:



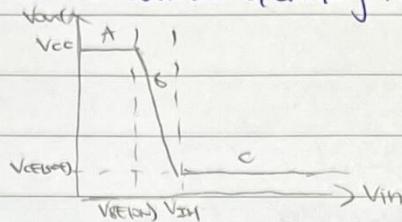
For Personal Use Only -bkwk2

The bipolar inverter (Bipolar transistor logic (RTL))

- The circuit diagram for the bipolar inverter is as follows:



- The voltage transfer characteristics can be obtained by analysing the different regimes of the BJT.



(i) Region A: $0 \leq V_{IN} < V_{BE(on)}$

→ BJT OFF

→ No current flows (since BJT is OFF)

(ii) Region B: $V_{BE(on)} \leq V_{IN} < V_{BE(sat)}$

→ BJT ON (forward active)

→ BJT starts to conduct, $I_C = \beta F I_B$

(iii) Region C: $V_{IN} \geq V_{BE(sat)}$

→ BJT ON (saturation)

→ Current saturates, i.e. I_C no longer increases w/ increasing I_B , $I_C < \beta F I_B$.

→ $V_{CE} = V_{CE(sat)}$

- V_{OH} is at cut-off region (Region A) → BJT OFF → Output pulled to V_{CC} → $V_{OH} = V_{CC}$

V_{OL} is at saturation region (Region C) → $I_C = I_{C(sat)}$ → Voltage drop across R_C const. → $V_{OL} = V_{CE(sat)}$

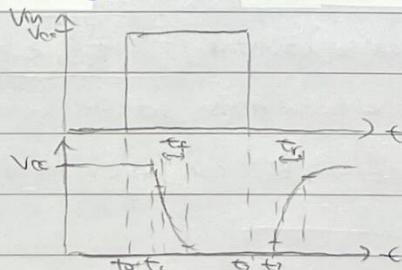
V_{IH} is at the edge of saturation → $V_{BE} = V_{BE(sat)}$, $I_C = \beta F I_B$ → $V_{IH} = V_{RE} + I_B R_B = V_{BE(sat)} + R_B \frac{V_{CC} - V_{BE(sat)}}{\beta F R_C}$

V_{IL} is at the edge of cutoff → $V_{BE} = V_{BE(on)}$ → $V_{IL} = V_{BE(on)}$

- As usual, $NM_H = V_{OH} - V_{IH}$, $NM_L = V_{IL} - V_{OL}$, logic swing = $V_{OH} - V_{OL}$

Bipolar inverter switching speed

- The switching delays of the bipolar transistor can be visualised as follows:



- fall time t_f : discharge of stray capacitance at V_{OL} ; rise time t_r : charging of stray capacitance through R_C

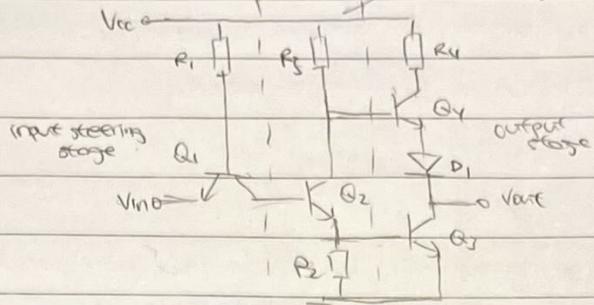
transistor switching delays, t_1-t_2 : charging of stray capacitance in base circuit

t_3 : discharging of stray capacitance in base circuit + removal of minority carriers from base region

For Personal Use Only -bkwk2

Transistor-transistor logic (TTL) inverter.

- The circuit diagram for the TTL inverter is as follows: phase splitting stage



- Q_1, R_1 form the i/p steering stage:

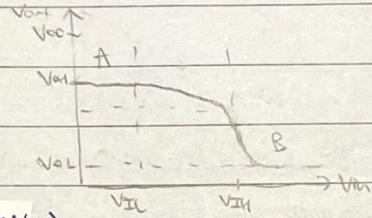
$\hookrightarrow V_{in} \text{ HIGH} \rightarrow B-E$ reverse-biased, $B-C$ forward-biased (reverse active) \rightarrow current flows to Q_2 through R_1

$\hookrightarrow V_{in} \text{ LOW} \rightarrow B-E$ forward-biased, $B-C$ reverse-biased (forward active) \rightarrow current flows to o/p through R_2

- Q_2 provides extra gain \rightarrow additional current to switch the o/p stage Q_3 more effectively

- Q_4 replaces the resistive load as an active pull-up. It has more gain and can source more current than a resistor. Q_3 and Q_4 form a totem-pole o/p stage.

- The voltage transfer characteristic is as follows:



(i) Region A: V_{in} LOW ($V_{in} < V_{IL}$)

$\hookrightarrow Q_1$ ON (forward active)

$\hookrightarrow Q_2$ OFF $\rightarrow V_{B3}$ pulled to GND, V_{B4} pulled to V_{cc} $\rightarrow Q_3$ OFF, Q_4 ON (saturation)

$$\hookrightarrow V_{out} = V_{cc} - R_4 I_4 - V_{CE(sat)} = V_{OL} = V_{OH}$$

(ii) Region B: V_{in} HIGH ($V_{in} > V_{IH}$)

$\hookrightarrow Q_1$ ON (reverse active)

$\hookrightarrow Q_2$ ON (saturation) $\rightarrow Q_3$ ON (saturation) $\rightarrow V_{E2} = V_{BE(sat)}$, $V_{C2} = V_{BE(sat)} + V_{CE(sat)}$, $V_{C3} = V_{CE(sat)}$

$$\hookrightarrow V_{B4} = V_{C2} = V_{BE(sat)} + V_{CE(sat)}, V_{B4} = V_{C3} + V_{BE(sat)} = V_{CE(sat)} + V_{BE(sat)} \rightarrow V_{BE4} = 0 < V_{BE(on)} \rightarrow Q_4$$
 OFF,

$$\hookrightarrow V_{out} = V_{C3} = V_{CE(sat)} = V_{OL}$$

- Typical values are $V_{BE(on)} = 0.65\text{V}$, $V_{FE(on)} = 0.75\text{V}$, $V_{CE(sat)} = 0.2\text{V}$

$$V_{OH} = 3.6\text{V}, V_{OL} = 0.2\text{V}, V_{IH} = 1.5\text{V}, V_{IL} = 0.6\text{V} \rightarrow NM_L = 0.4\text{V}, NM_H = 2.1\text{V}, \log_2 \text{swing} = 2.4\text{V}$$

- TTL uses high conductance paths to drive the base \rightarrow lower propagation delays.

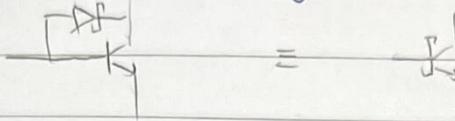
- V_{out} for TTL is significantly lower than V_{cc} , and is still relatively slow as we need to remove minority carriers when bringing the BJT out of saturation

* For gates w/ multiple i/p's, we can replace Q_1 w/ a transistor w/ two emitters,

For Personal Use Only -bkwk2

Schottky TTL (STL)

- In a schottky transistor, a Schottky Diode is connected across the base-collector junction of each bipolar transistor to anti-alias (prevent junction from being forward biased by noise)
- Avoiding saturation allows for faster switching



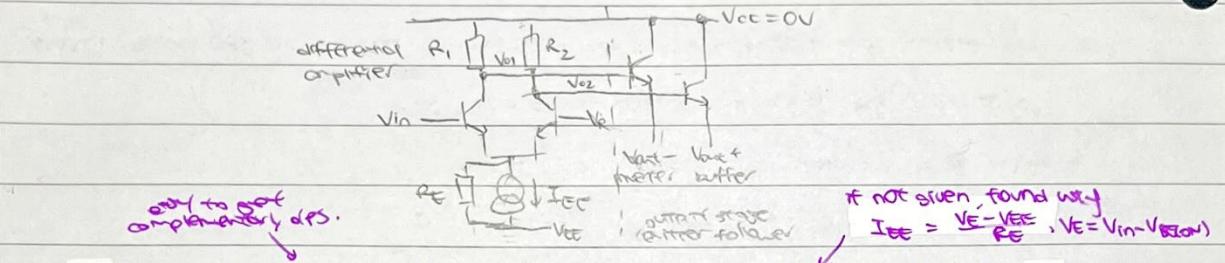
delay power consumption
trade off.

- STL has switching times $\approx 3\text{ns}$ and has twice the power consumption of TTL

Low-power STL uses high value resistors $\rightarrow 10\times$ less power consumption, 3x slower switching.

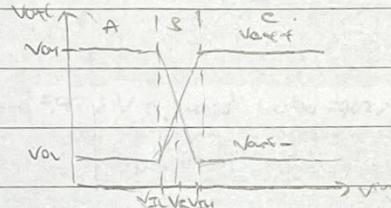
Emitter-coupled logic (ECL) inverter

- The circuit diagram for a ECL inverter is as follows:



- The top stage is a differential amplifier w/ a constant tail current IEE . Vin rel to Vc steers IEE b/w the two branches \rightarrow act as non-saturating current switch.

- The voltage-transfer characteristic is as follows:



(i) Region A: $Vin < VR$

$\hookrightarrow Q_1 \text{ OFF}, Q_2 \text{ ON (forward-active)}$

$\hookrightarrow V_{O1} = V_{CC}, V_{O2} = V_{CC} - I_{EE} R_2$ [ideal case: $I_{E1} = 0, I_{E2} = I_{EE}$]

(ii) Region B: $Vin = VR$

$\hookrightarrow Q_1 \text{ I}, Q_2 \text{ ON (forward-active)}$

$\hookrightarrow V_{O1} = V_{CC} - (\frac{I_{EE}}{2})R_1; V_{O2} = V_{CC} - (\frac{I_{EE}}{2})R_2$

(iii) Region C: $Vin > VR$

$\hookrightarrow Q_1 \text{ ON (forward-active)}, Q_2 \text{ OFF}$

$\hookrightarrow V_{O1} = V_{CC} - I_{EE} R_1, V_{O2} = V_{CC}$ [ideal case: $I_{E1} = I_{EE}, I_{E2} = 0$]

$\hookrightarrow V_{out-} = V_{O1} - V_{BE(\text{ON})}$ and $V_{out+} = V_{O2} - V_{BE(\text{ON})}$

$\hookrightarrow V_R$ can be found using regular circuit analysis. To simplify calculations, we can assume base currents of transistors to be negligible, and all diodes / base-emitter junctions have $\Delta V = V_{BE(\text{ON})}$

(typically, find the base voltage of the BJT stage before V_R , then use $V_E = V_B - V_{BE(\text{ON})}$)

For Personal Use Only -bkwk2

- The transition region is defined in terms of emitter current of Q1/Q2 or the change in i/p voltage when switching the emitter current from 5% to 95% of its final value.

- Note in practice, we cannot steer I_{EE} if it only flows through one branch $\rightarrow I_{EE} = I_{E1} + I_{E2}$.

$$\frac{I_{E1}}{I_{E2}} = \frac{I_{E0} e^{(V_{B1}-V_E)/V_T}}{I_{E0} e^{(V_{B2}-V_E)/V_T}} = e^{(V_{B1}-V_E)/V_T} \rightarrow V_{B1} = V_R - V_T \ln\left(\frac{I_{E1}}{I_{E2}}\right)$$

$$\text{At } I_{E1} = 0.05 I_E, \quad V_{B1} = V_{IL} = V_R - V_T \ln\left(\frac{0.75}{0.05}\right) ; \text{ At } I_{E1} = 0.95 I_E, \quad V_{B1} = V_{IH} = V_R - V_T \ln\left(\frac{0.95}{0.05}\right)$$

$$\Delta V_1 = V_{IH} - V_{IL} = 2V_T \ln 19 \approx 150 \text{ mV}, \text{ taking } V_T = 25 \text{ mV}.$$

- Typical values are $V_{EE} = -5.2V$, $V_R = -1.3V$, $I_{EE} = 2 \text{ mA}$, $V_{Emin} = 0.65V$

$$V_{OH} = -0.9V, V_{OL} = -1.75V, V_{IH} = -1.225V, V_{IL} = -1.375 \rightarrow NM_L = 0.375V, NM_H = 0.225V, \text{ logic supply} = 0.25V.$$

- ECL is the fastest logic circuit available conventionally. (BJT never in saturation \rightarrow fast)

- Differential nature of circuit makes it resilient to noise and complementary clfs are available.

- Current drawn from supply remains const. during switching (unlike CMOS or TTL)

- Note the o/p voltage levels are not compatible w/ other logic families \rightarrow hard to integrate.

Types of ECL

- Two of the most popular forms of commercially available ECL is ECL 10K and ECL 100K.

↳ ECL 10K : lower power dissipation but is slow [gate delay = 2ns, power dissipation = 25mW]

↳ ECL 100K : faster switching but higher power dissipation [gate delay = 0.75ns, power dissipation = 40mW]

- Other forms of ECL include PECL and LVPECL

↳ Positive ECL (PECL) : $V_{CC} = 5V, V_{EE} = 0V \rightarrow$ easier to interface w/ other logic families.

(However, logic levels are defined rel. to supply rail, diff. to other logic families)

↳ Low Voltage PECL (LVPECL) : $V_{CC} = 3.1V, V_{EE} = 0V \rightarrow$ power optimised version of PECL.

- The voltage levels used in ECL, PECL and LVPECL are summarised below:

| Type | V_{EE} | V_{OL} | V_{OH} | V_{CC} |
|--------|----------|----------|----------|----------|
| ECL | -5.2V | -1.75V | -0.9V | GND |
| PECL | GND | 3.4V | 4.2V | 5V |
| LVPECL | GND | 1.0V | 2.9V | 3.3V |

For Personal Use Only -bkwk2

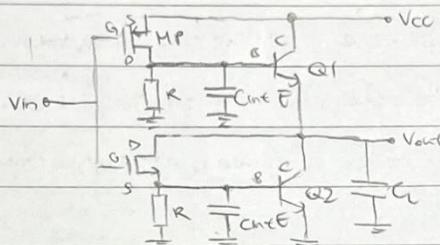
BiCMOS transistor and logic circuits

BiCMOS Transistor

- CMOS has low static power consumption, high i/p impedance and wide noise margins.
- BJTs have a large current o/p \rightarrow high f/out
- We can multiply the o/p of MOSFET using BJTs to form a BiCMOS transistor.
- BiCMOS transistors have high i/p impedance, slightly worse noise margins but much better f/out.

BiCMOS inverter

- The circuit diagram for the BiCMOS inverter is as follows:



(i) $V_{in} > V_{IH}$ ($V_{in} + I_{GTH}$)

\hookrightarrow MN is ON \rightarrow base and collector of Q2 shorted

\hookrightarrow Q2 starts conducting \rightarrow Vout pulled to GND \rightarrow C_L discharges through Q2 ($V_{out} = V_{BE2}$)

(ii) $V_{in} < V_{IL}$ ($V_{in} - I_{GTL}$)

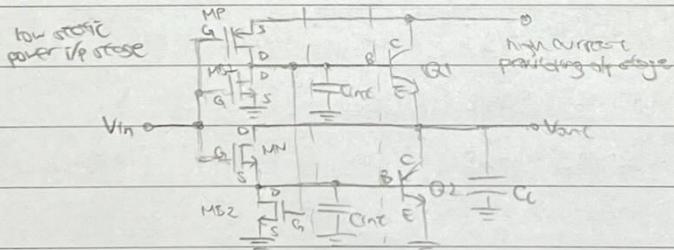
\hookrightarrow MP is ON \rightarrow base and collector of Q1 shorted

\hookrightarrow Q1 starts conducting \rightarrow Vout pulled to Vcc \rightarrow C_L charges through Q1 ($V_{out} = V_{CC} - V_{BE1}$)

\star The base charges of Q1/Q2 (represented by C_{int}) discharge through the resistors.

BiCMOS inverter with active pullup pull-down

- The circuit diagram for the BiCMOS inverter w/ active pullup pull-down is as follows:



\star Resistors increase power consumption and req. large surface area \rightarrow expensive \rightarrow replace R w/ M81, M82

\star For $V_{in} \downarrow V_{IL}$, MP ON, M81 OFF / For $V_{in} \uparrow V_{IH}$, MN ON, M82 OFF \rightarrow all current charge C_{int}

When MP is OFF, M81 is ON / When MN is OFF, M82 is ON \rightarrow C_{int} can discharge through M81 (M82)

\star Operation essentially the same as regular BiCMOS inverter.

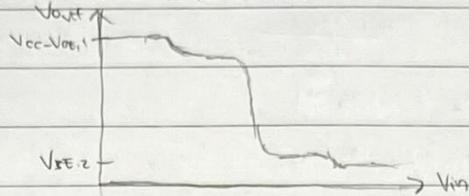
\star Current o/p's of MOS transistors are low \rightarrow use larger $\frac{W}{L}$ (or use cascaded stages w/ increasing $\frac{W}{L}$)

Current is amplified at the o/p stage by BJTs (who increases the overall static power consumption much)

For Personal Use Only -bkwk2

Performance of RCMOS inverter

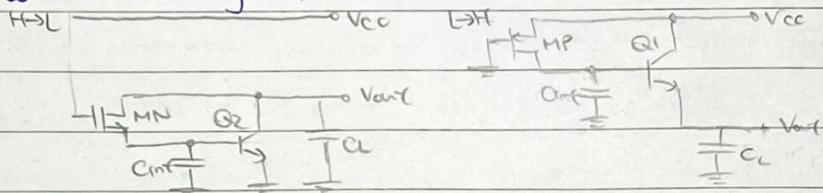
- The voltage transfer characteristic is as follows:



- V_{out} is upper bounded by $V_{cc} - V_{ce,1}$ and lower bounded by $V_{ee,2} \rightarrow$ slightly worse V_{out} , V_{in} performance
sharp fall \rightarrow better inverter performance (good logic swing and large margins)

- Both $V_{ce,1}$ and $V_{ce,2}$ are above saturation levels $\rightarrow Q1(Q2)$ never enter saturation \rightarrow faster switching
- It is hard to fully analyze propagation delays in RCMOS (many devices involved), so assume the following:
 - \hookrightarrow i/p signal switching occurs instantaneously
 - \hookrightarrow load is purely capacitive
 - \hookrightarrow approximate MN, MP as resistors when conducting

- We will only consider the following simplified circuits



① TTL propagation delay t_{phl}

\hookrightarrow Once ON, replace MN w/ equivalent resistor R_N , $R_N = \frac{V_{cc}}{2k_N(V_{cc}-V_{tn})^2}$

\hookrightarrow Propagation delay consists of (i) charging of C_{int} and (ii) discharging of C_L

(i) C_{int} needs to be charged up to $V_{ce(on)}$ to make Q2 ON

$$t_1 = \frac{Q}{C_{int}} = \frac{C_{int} V_{ce(on)}}{(V_{cc} - V_{ce(on)}) / R_N} = \frac{C_{int} R_N V_{ce(on)}}{V_{cc} - 0.5 V_{ce(on)}}$$

(ii) Q2 ON $\rightarrow C_L$ discharged to $V_{cc}/2$ by Q2

total capacitance at base side: $C_T = C_{int} + \frac{C_L}{\beta+1}$ capacitance referred from emitter to base

$$t_2 = \frac{Q}{C_T} = \frac{C_T V_{cc}/2}{V_{cc}/2 R_N} = \frac{(C_{int} + C_L/\beta+1) V_{cc}/2}{V_{cc}/2 R_N} = (C_{int} + \frac{C_L}{\beta+1}) R_N$$

$$\Rightarrow t_{phl} = t_1 + t_2 = \frac{C_{int} R_N V_{ce(on)}}{V_{cc} - 0.5 V_{ce(on)}} + (C_{int} + \frac{C_L}{\beta+1}) R_N$$

② L-T propagation delay t_{plh}

\hookrightarrow Once ON, replace MP w/ equivalent resistor R_P , $R_P = \frac{V_{cc}}{2k_P(V_{cc}-V_{tp})^2}$

\hookrightarrow Propagation delay consists of (i) charging of C_{int} and (ii) charging of C_L

(i) C_{int} needs to be charged up to $V_{ce(on)}$ to make Q1 ON.

$$t_1 = \frac{Q}{C_{int}} = \frac{C_{int} V_{ce(on)}}{(V_{cc} - V_{ce(on)}) / R_P} = \frac{C_{int} R_P V_{ce(on)}}{V_{cc} - 0.5 V_{ce(on)}}$$

(ii) Q1 ON $\rightarrow C_L$ charged to $V_{cc}/2$ by Q1

total capacitance at base side: $C_T = C_{int} + \frac{C_L}{\beta+1}$

$$t_2 = \frac{Q}{C_T} = \frac{C_T V_{cc}/2}{V_{cc}/2 R_P} = \frac{(C_{int} + C_L/\beta+1) V_{cc}/2}{V_{cc}/2 R_P} = (C_{int} + \frac{C_L}{\beta+1}) R_P$$

$$\Rightarrow t_{plh} = t_1 + t_2 = \frac{C_{int} R_P V_{ce(on)}}{V_{cc} - 0.5 V_{ce(on)}} + (C_{int} + \frac{C_L}{\beta+1}) R_P$$

For Personal Use Only -bkwk2

Comparison of CMOS and BiCMOS.

- For BiCMOS, $t_{PLH} = \frac{C_{int} R_p V_{DDQV}}{V_{ce} - 0.5 V_{BEQV}} + (C_{int} + \frac{C_L}{\beta T_1}) R_p = \alpha C_{int} + \frac{b}{\beta T_1} C_L$, where $\alpha = 1.2 R_p$, $b = R_p$

$$t_{PLH} = \frac{C_{int} R_p V_{DDQV}}{V_{ce} - 0.5 V_{BEQV}} + (C_{int} + \frac{C_L}{\beta T_1}) R_N = c C_{int} + \frac{d}{\beta T_1} C_L, \text{ where } c = 1.2 R_N, d = R_N$$

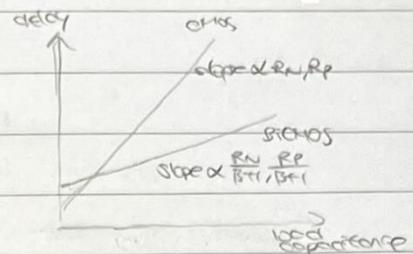
- For CMOS, $t_{PLH} = 0.7 R_p C_L$

$$t_{PLH} = 0.7 R_N C_L$$

→ For low C_L , coeff a, c dominant → CMOS performs better

For high C_L , coeff b, d dominate → BiCMOS performs better

→ BiCMOS manufacturing process is more complex → greater cost per gate.



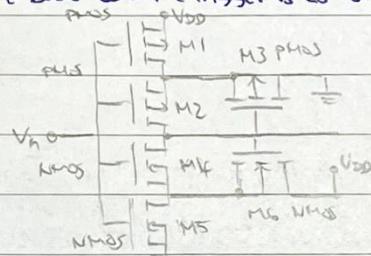
CMOS Schmitt trigger

CMOS Schmitt trigger

- Schmitt trigger have hysteresis — H-L op and L-H op transitions occur at diff. i/p voltage.

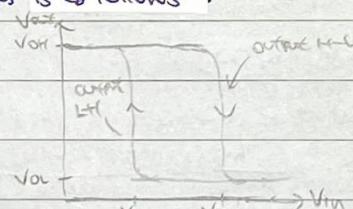
- Schmitt trigger are useful for speeding up slowly changing signals and cleaning up noisy signals.

- The circuit diagram for the basic schmitt trigger is as follows:



M3, M6 affect only FB signals from the o/p back to the series connected pair of transistors.

- The voltage transfer characteristic is as follows:



To determine the critical pds, assume that no current is drawn from the o/p, V_{out} .

(i) V_{out} HIGH, assume $V_{in} = 0V$

→ V_{in} at GND → M5 OFF → no pull-down path to GND

→ V_{in} at GND → M1 ON (non-saturation) → V_{S1} pulled to virtual V_{DD} → M2 ON (non-saturation)

→ High conductance path between V_{DD} and V_{out} → $V_{out} = V_{DD}$

(ii) V_{out} LOW, assume $V_{in} = V_{DD}$

→ V_{in} at V_{DD} → M1 OFF → no pull-up path to V_{DD}

→ V_{in} at V_{DD} → M5 ON (non-saturation) → V_{S5} pulled to virtual GND → M4 ON (non-saturation)

→ High conductance path between GND and V_{out} → $V_{out} = 0V$

→ logic swing = V_{DD} (good)

For Personal Use Only -bkwk2

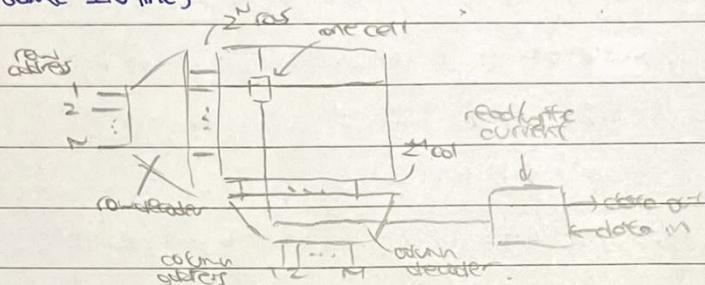
semiconductor memories

semiconductor memories

Memory elements

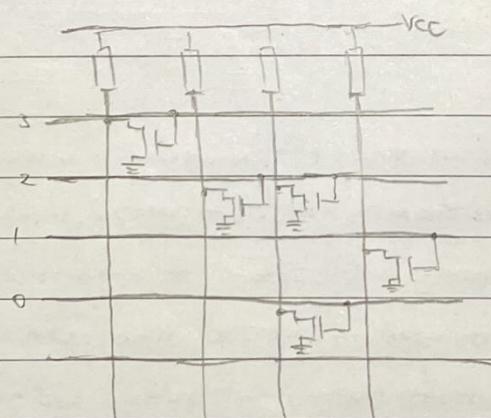
- A device having more than one bit of addressing req. addressing, where an address designates a specific storage location. The bits of the device are the values stored in these locations.
- There are three modules in a memory device.
 - ↳ 1) Memory array - $N \times M$ array of one bit storage cells (total memory is $2^L \times 2^N = 2^{M+N}$)
 - ↳ 2) Row decoder - For given N bit address i/p, it selects a unique horizontal line out of 2^N lines.
(Each selection corresponds to a word, stored in the corresponding row of the memory array)
 - ↳ 3) Column decoder - For given M bit i/p, selects from 2^M vertical lines for read/write operation.
(Selects the bit inside the word selected by the row decoder).

- ↳ Only one row can be addressed at a time, or data becomes indeterminate (since cells on the same column share the same I/O line)



Read only memory (ROM)

- A ROM's content is written only once (i.e. read only).
- Data stored is encoded by absence or presence of MOS transistors at row-column intersections.
(Gate connected w/ address i/p, drain connected to columns, source grounded).
- Upon address selection, transistors in the corresponding line are turned ON
 - ↳ If there is a transistor, it is turned ON \rightarrow drain is pulled to GND \rightarrow logic 0 is read
 - ↳ If there is no transistor, the column i/e remains intact \rightarrow logic 1 is read

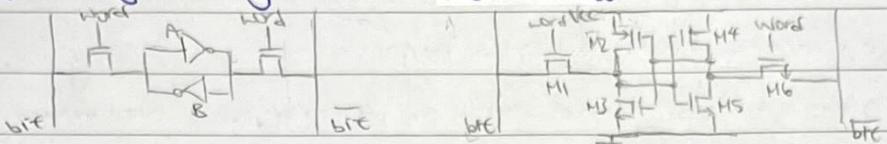


For Personal Use Only -bkwk2

static random access memory (SRAM).

- For SRAM, it is necessary to read out their data and also update/write to data.

- A basic RAM cell consists of a pair of cross-coupled MOS inverters. The inverted can be accessed through switching transistors, which are transistors connected to the row address lines.



- The RAM cell can be either logic 1 or logic 0.

↳ Inverter B is generating logic 1/0 → inverter A inverts logic 0/1 → inverter B

outputs logic 1/0, i.e. FB turns the inverters, state is kept intact.

- To change the state of the RAM cell,

↳ Drive the vertical bitlines to the new values

↳ Operate the address lines to switch on the pass transistors - connect cell to bit lines.

* The RAM cell inverters must be sufficiently "weak" so the RD-line signals can override the current op values - use M2-M5 width $\frac{W}{2}$, M1,M6 width $\frac{W}{2}$

- To read the state of the RAM cell,

↳ Both bit lines are preset to precisely the same voltage level

↳ Pass transistors activated → inverter outputs reflect the bit lines by a few mV → e.g. disturbed (extra circuitry added to speed up this process)

↳ After reading, the pass transistors are turned off → RAM cell settles back to original value.

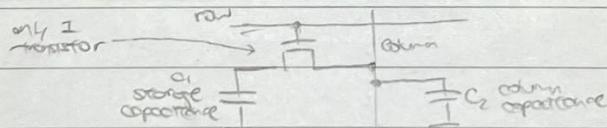
Dynamic random access memory (DRAM)

DRAM cheaper than SRAM
when memory > 256kB.

- DRAM uses 6 transistors per bit and several control lines → high cost of memory per bit.

DRAM solves this by using memory cells based on a capacitive storage element (reduced component count)

- Logic 0/1 is represented by the presence/absence of a small pocket of charged stored in a capacitor.



- Leaking currents can remove the stored charge in a few ms → necessary to refresh/reload the logic 0 signal more frequently than this → additional complexity in the control/drive circuitry.

- Read/write by applying a logic 1 to the gate of the transistor via the row

↳ Data written into the cell by logic 1/0 on the column line while the cell is selected.

↳ When reading, charge stored on C1 is shared w/ much larger capacitance C2 of the column line

→ we measure the change in potential (may be susceptible to noise) → use differential sensing (two FB)

* $Q_{before} = C_1 V_{in} + C_2 V_{in}$, $Q_{after} = (C_1 + C_2)(V_{out} + \Delta V)$ → solve to find ΔV when logic 1/0 stored in C1

we can solve for V_{in} given a sensing threshold ΔV_{th} → find how freq. we need to refresh, $f = \frac{1}{T_{decay}}$
Power dissipated found using $P = \frac{1}{2} C_1 (V_f^2 - V_i^2)$, where $V_f = V_i - \frac{I_{leak} T_r}{C_1}$

For Personal Use Only -bkwk2

(iii) Output H-L transition (V_{in} rises from GND)

↳ V_{in} rises above V_{TN} → M5 turns ON (saturation)

↳ $V_{D,6}$ still at $V_{out} = V_{DD} = V_{DD}$. since $V_{D,6} = V_{DD} \rightarrow V_{DS,6} = V_{GS,6} > V_{GS,6} - V_{TN} \rightarrow M_6$ saturated.

↳ while M4 remains non-conductive, equating drain currents $I_{D,5}, I_D$,

$$I_{D,5(\text{sat})} = I_{D,6(\text{sat})} \rightarrow \frac{k_F}{2}(V_{in} - V_{TN})^2 = \frac{k_F}{2}(V_{GS,6} - V_{TN})^2 \rightarrow V_{GS,6} = \sqrt{\frac{k_F}{k_B}}(V_{in} - V_{TN}) + V_{TN}$$

↳ Noting that $V_{D,5} = V_{DD} - V_{GS,5}$, we have $V_{D,5} = V_{DD} - \sqrt{\frac{k_F}{k_B}}(V_{in} - V_{TN}) - V_{TN}$

↳ M4 first turns on when $V_{GS,4} = V_{in} - V_{D,5} = V_{TN}$. At this pt., $V_{in} = V_{ID}$

$$V_{ID} = V_{D,5} + V_{TN} = V_{DD} - \sqrt{\frac{k_F}{k_B}}(V_{ID} - V_{TN}) \rightarrow V_{ID} = \frac{V_{DD} + \sqrt{\frac{k_F}{k_B}}V_{TN}}{1 + \sqrt{\frac{k_F}{k_B}}}$$

(iv) Output L-H transition (V_{in} falls from V_{DD})

↳ V_{in} falls below $V_{DD} - |V_{tr}| \rightarrow M_1$ turns ON (saturation)

↳ $V_{G,3}$ still at $V_{out} = V_{DD} = 0V$. since $V_{D,2} = 0V \rightarrow V_{D,2} = V_{GS,3} > V_{GS,3} - |V_{tr}| \rightarrow M_3$ saturated

↳ while M2 remains nonconductive, equating drain currents $I_{D,1}, I_{D,5}$,

$$I_{D,1(\text{sat})} = I_{D,5(\text{sat})} \rightarrow \frac{k_F}{2}(V_{DD} - V_{in} - |V_{tr}|)^2 = \frac{k_F}{2}(|V_{GS,1}| - |V_{tr}|)^2 \rightarrow |V_{GS,1}| = \sqrt{\frac{k_F}{k_B}}(V_{DD} - V_{in} - |V_{tr}|) + |V_{tr}|$$

↳ Noting that $V_{D,1} = 0 + |V_{GS,1}|$, we have $V_{D,1} = \sqrt{\frac{k_F}{k_B}}(V_{DD} - V_{in} - |V_{tr}|) + |V_{tr}|$

↳ M2 first turns on when $|V_{GS,2}| = V_{D,1} - V_{in} = |V_{tr}|$. At this pt., $V_{in} = V_{IU}$

$$V_{IU} = V_{D,1} - |V_{tr}| = \sqrt{\frac{k_F}{k_B}}(V_{DD} - V_{in} - |V_{tr}|) \rightarrow V_{IU} = \frac{\sqrt{\frac{k_F}{k_B}}(V_{DD} - |V_{tr}|)}{1 + \sqrt{\frac{k_F}{k_B}}}$$

↳ V_{ID} is the opo H-L trip voltage and V_{IU} is the opo L-H trip voltage, $V_{IL} > V_{IU}$

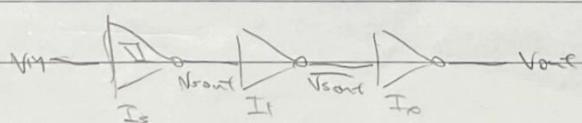
CMOS Schmitt trigger device dimensions

- V_{ID} and V_{IU} depend on the process gain factor k of the FB devices (M3, M4), which in turn depend on the $(\frac{W}{L})$ of the corresponding transistors.

- Increasing the $(\frac{W}{L})$ ratio for M3, M4 results in greater diff. b/w V_{ID}, V_{IU} and the mid-pt voltage $\frac{V_{DD}}{2}$, giving greater hysteresis, and vice versa.

Buffered schmitt trigger

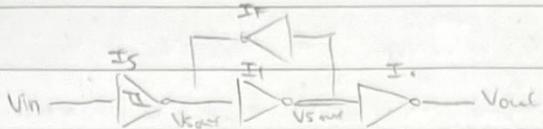
- The i/p stage of CMOS schmitt inverter consists of 4 MOS transistors in series. For comparable current, w/l CMOS inverter (2 MOS transistors), each device must have double the $(\frac{W}{L})$ ratio
- the increase in size of the devices result in
 - ↳ increase the i/p capacitance to an unmanageable amount
 - ↳ decrease in resistance $R_N \propto (\frac{W}{L})$
- If we req. a CMOS schmitt inverter w/ high drive capability is req. it is best to use the smaller possible devices in the i/p stage (w/l, capacitance), followed by a two-stage inverting buffer.



For Personal Use Only -bkwk2

Schmitt trigger with regenerative feedback.

- A FB inverter If can improve the transient response of a CMOS Schmitt trigger w noisy i/p.



- If provides further noise immunity by operating the FB \rightarrow op switches more sharply (positively).

- To avoid contention, forward inverter If should have more drive capability than If.

(We can achieve this by using transistors of lower K for If)

For Personal Use Only -bkwk2

Combinational and sequential circuits

Combinational circuits

Boolean algebra

- The theorems of switching algebra are as follows:

Commutation

$$a+b = b+a$$

$$ab = ba$$

Association

$$(a+b)+c = a+(b+c)$$

$$(ab)c = a(bc)$$

Distribution

$$a(b+c) = ab+ac$$

$$a+(bc) = (a+b)(a+c)$$

Absorption

$$a+ab = a$$

$$a(a+b) = a$$

De-Morgan

$$\overline{a+b} = \overline{a}\overline{b}$$

$$(a+b)(a+\overline{b}) = a$$

SOP, POS and canonical form

- The SOP is made of minterms (m) and the POS is made of maxterms (M).

There are 2^n minterms and maxterms (n is the no. of variables).

→ e.g. FOR variables a, b, c ($n=3$), $M_0 = \overline{a}\overline{b}\overline{c}$, $M_1 = \overline{a}\overline{b}c$... $M_7 = abc$

$$M_0 = ab+bc, M_1 = ab+\overline{c} \dots M_7 = \overline{a}+\overline{b}+\overline{c}$$

- In canonical form, all the binary variables appear once in each term (SOP) or factor (POS)

To expand an expression into its canonical form, we use $x+x=1 \rightarrow$ we get BCD $f=\sum(l)$.

- For SOP, expand the BCD $f=\sum(l)$

or get via K-map

For POS, find BCD for \bar{f} , $\bar{f}=\sum(l)$, expand, then apply De-Morgan's.

- * Canonical form is useful for MUX, ROM, LUT; minimized form is useful w/ using min. no. of logic gates.

Karnaugh maps.

- Karnaugh maps for 3/4/5 variables are shown below.

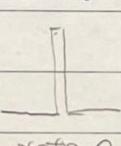
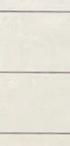
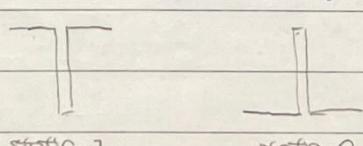
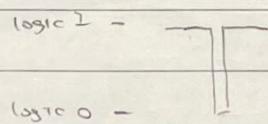
| | | | | | | | | a | | a | | a | | a | |
|---|---|---|---|---|---|---|----|----|---|----|----|----|----|----|----|
| | | | | | | | | 0 | 8 | 16 | 24 | 32 | 40 | 48 | 56 |
| | | | | 0 | 2 | 6 | 4 | 1 | 5 | 13 | 9 | 2 | 10 | 25 | 18 |
| c | 1 | 3 | 7 | 5 | 3 | 7 | 15 | 11 | 6 | 14 | 20 | 22 | 7 | 15 | 31 |
| b | | | | | 2 | 6 | 14 | 10 | 4 | 12 | 22 | 20 | 5 | 13 | 29 |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

- we can easily find the BCD of a function f or its inversion \bar{f} using a K-map.

(use $f=\sum(l)$ for SOP/NAND implementation, $\bar{f}=\sum(l)$ for POS/NOR implementation)

- we can remove static hazards by ensuring all product terms overlap in the K-map.

(K-map for f to remove static-1 hazard; K-map for \bar{f} to remove static-0 hazard).

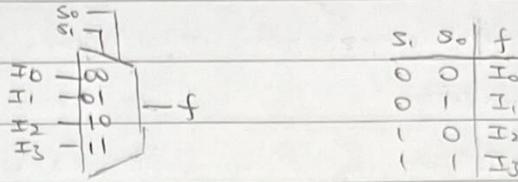


For Personal Use Only -bkwk2

Multiplexers (MUX)

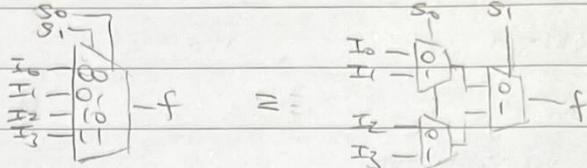
- A MUX uses the select i/p's to select which data i/p to output.

Eg, for a 4-to-1 MUX,



$$\text{SOP implementation: } f = \bar{S}_1 \bar{S}_0 I_0 + \bar{S}_1 S_0 I_1 + S_1 \bar{S}_0 I_2 + S_1 S_0 I_3$$

- Larger MUXs can be constructed from smaller ones (smallest is 2-to-1 MUX)



- MUX can be used to realise combinational circuits

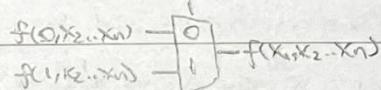
↳ Using ext. i/p's only as select i/p's \rightarrow LUT.

↳ Using some ext. i/p's as select i/p's \rightarrow Shannon's expansion thm.

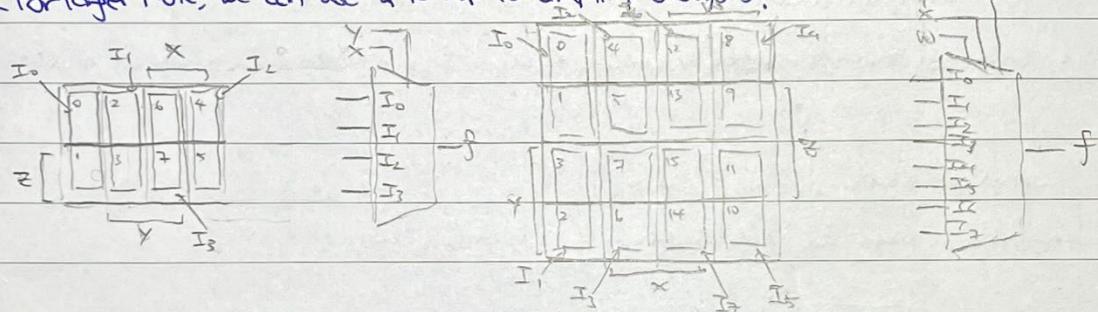
- Shannon's expansion thm. states any boolean function $f(x_1, x_2, \dots, x_n)$ can be written in the form

$$f(x_1, x_2, \dots, x_n) = \bar{x}_1 f(0, x_2, \dots, x_n) + x_1 f(1, x_2, \dots, x_n)$$

We can use a 2-in-1 MUX w/ x_i as select i/p.



- For larger MUX, we can use a K-map to simplify analysis.

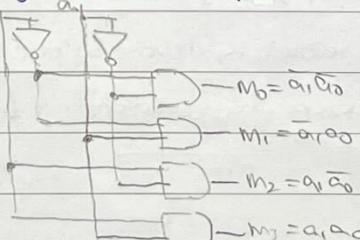


The data i/p's are functions of Z . This can be found using a truth table

| $I=0$ | $I=1$ | $I=2$ | $I=3$ |
|-------|-------|-------|-------|
| $Z I$ | $Z I$ | $Z I$ | $Z I$ |
| 0 0 | 0 0 | 0 1 | 0 1 |
| 1 0 | 1 1 | 1 0 | 1 1 |

Decoders

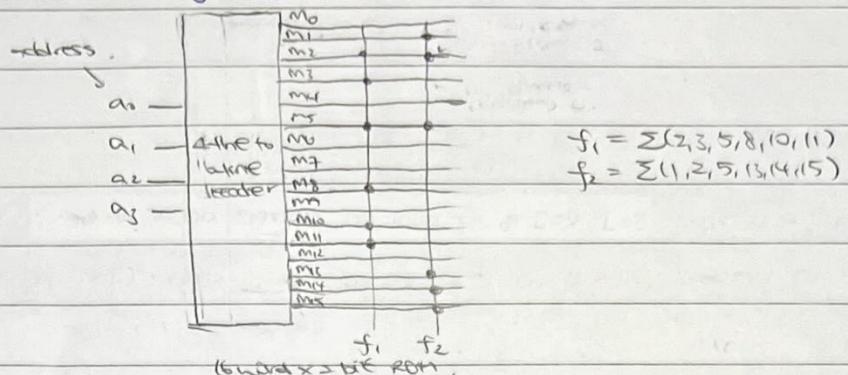
- A decoder takes n i/p's and 2^n o/p's \rightarrow asserts exactly one of 2^n o/p's depending on the i/p combination.



For Personal Use Only -bkwk2

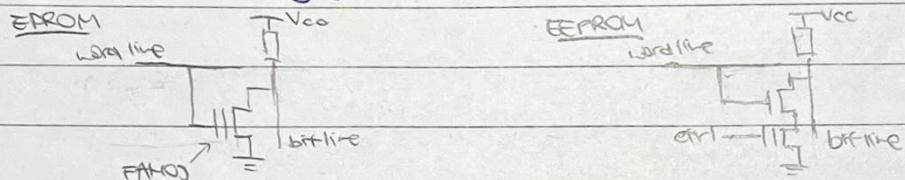
Read only memory (ROM)

- ROM is an interconnected array of semiconductor devices to store an array of binary data.
- stored data cannot be changed under normal operation conditions.



- in the ROM matrix indicates a programmable link - equivalent to storing a "1".
- Each cell in the ROM is implemented using floating gate transistors (FATOS) (FLOTOX).
- ↳ Floating gate transistors are programmed "1" by applying a high voltage \rightarrow e⁻ trapped on the floating gate
 $\rightarrow V_{th} \text{ shift} \rightarrow$ transistor never conducts during normal operation (1)
- ↳ If the floating gate transistor is not programmed "0", it acts like a regular NMOS transistor
 \rightarrow word line high \rightarrow transistor conducts, providing a path to GND (0).

- Common types of floating gate transistors are FATOS and FETOX, used in EEPROM and EEPROM respectively.



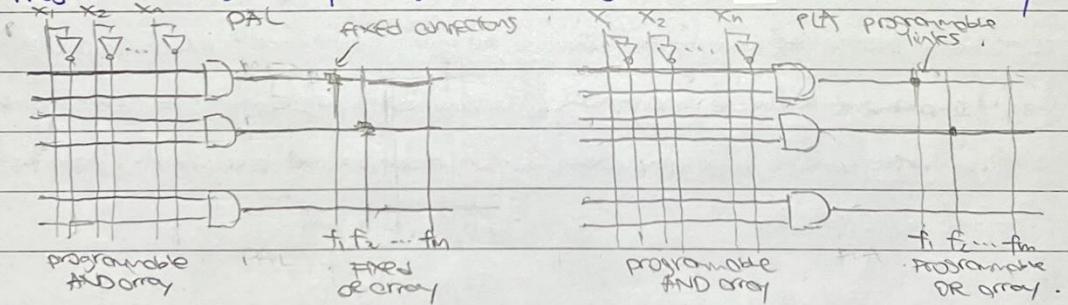
To program a "1", EEPROM: Gate and drain $\sim 20V$; EEPROM: Drain Low, Ctrl $\sim 2V$

To erase "1", EEPROM: UV light; EEPROM: Drain $\sim 2V$, Ctrl Low.

Programmable logic

- Programmable array logic (PAL) has a programmable AND array and fixed OR array.

Programmable logic array (PLA) has both programmable AND array and OR array



- + ROM has a fixed AND array and a programmable OR array.

For Personal Use Only -bkwk2

Sequential logic

Bistable circuits

- The excitation table for bistable circuits is as follows:

| OUTPUT $Q \rightarrow Q'$ | REQUIRED STATE | | | |
|------------------------------|----------------|-----|---|---|
| | S R | J K | T | D |
| 0 → 0 | 0 X | 0 X | 0 | 0 |
| 0 → 1 | 1 0 | 1 X | 1 | 1 |
| 1 → 0 | 0 1 | X 1 | 1 | 0 |
| 1 → 1 | X 0 | X 0 | 0 | 1 |

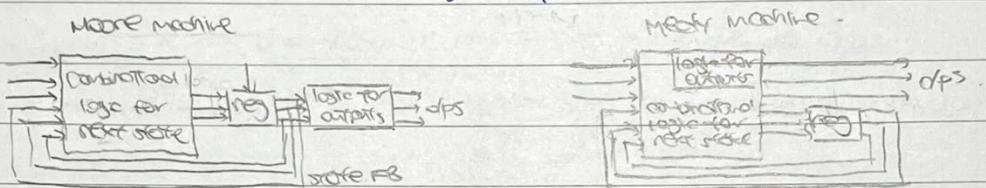
→ For SR bistable, changes occur at the rising clock edge ($S=1, R=1$ is not allowed)

For JK bistable, changes occur at the falling clock edge ($J=1, K=1$ toggles the d/p)

Models for sequential circuits.

- For a Moore machine, the o/p's are a function of the current state, $Z=f(Q)$, $Q' = f(x, Q)$
the o/p changes synchronously w/ state changes.
- For a Mealy machine, the o/p's depend on state and on i/p's, $Z=f(x, Q)$, $Q' = f(x, Q)$

i/p changes can cause immediate o/p changes (asynchronous)



Design of synchronous circuits

- Sequential circuits use a clock to synchronise all bistables' operation. The design steps are:

- ↳ 1) Design the problem and draw the state diagram
- ↳ 2) Draw the state table, remove redundant states and allocate bistables
- ↳ 3) Draw the state transition table (using excitation table for bistable circuits)
- ↳ 4) Draw a k-map for each bistable and the o/p.
- ↳ 5) Draw the circuit diagram

- To merge states, the next state must be the same / don't care for each i/p column.

However, they need not have the same d/p (though extra logic is req - Mealy machine)

- If we order the i/p's using gray code, we can read-off the k-maps entries from the state transition table.

For Personal Use Only -bkwk2

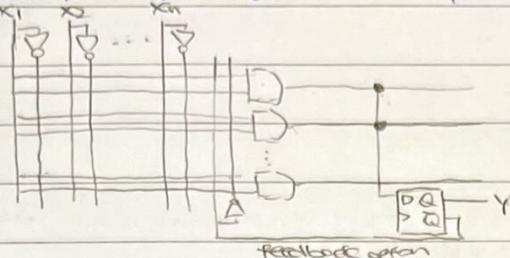
Programmable logic circuits, data storage processing and control

AN

Programmable logic circuits.

Sequential PAL.

- We can feed the PAL output through a register for memory or synchronisation — sequential PAL



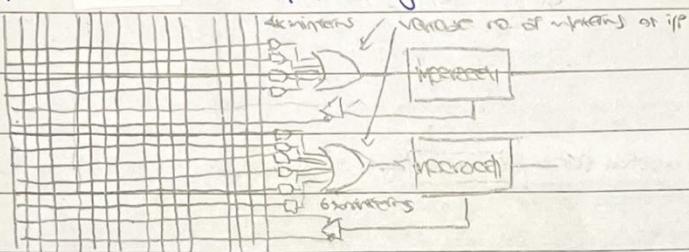
- Advantages of PAL:

- ↳ Fewer devices req. → smaller footprint, lower cost, power savings, easier to test + debug.
- ↳ High design flexibility, in system reprogrammability (some technology w/ EEPROM)

CY

Programmable logic device (PLD)

- A PLD is formed by one or multiple PAL arrays in a single device

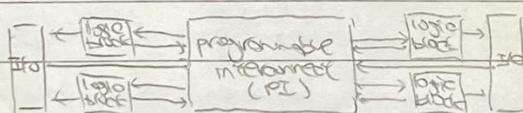


- Each macrocell is programmable and its function can vary depending on the MUX select pins.
It can switch between combinational/sequential circuits or input/output select.

CM

Complex PLD (CPLD)

- A CPLD is formed by combining multiple logic array blocks (LAB) in a single device w/ programmable interconnect and I/O (control & pins)



- Each LAB is effectively a PLD, but instead of variable no. of minterms as in gate IP, makes use of local programmable interconnect and expandable product terms.

- Note that the expandable product terms have an extra delay, but this is deterministic and can be accounted for in timing simulations.

- The PI is a global routing that connects any signal to any destination in device (programmed w/ EEPROM)

- The I/O control blocks are separated from LAB by PI. Tri-state buffer control is used to enable I/Os or bidirectional on any I/O pin. (In PLD, I/O connected directly to logic)

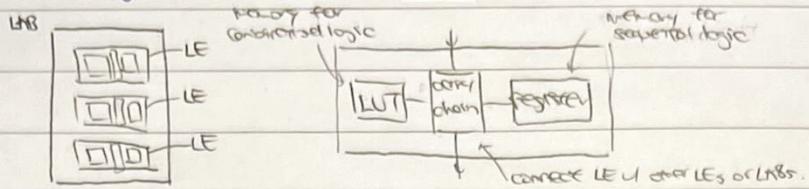
- Advantages of CPLD:

- ↳ High flexibility in logic and routing (greater capability than PLDs) reprogrammable.
- ↳ Instantly turn on or board power up (some for PLDs)

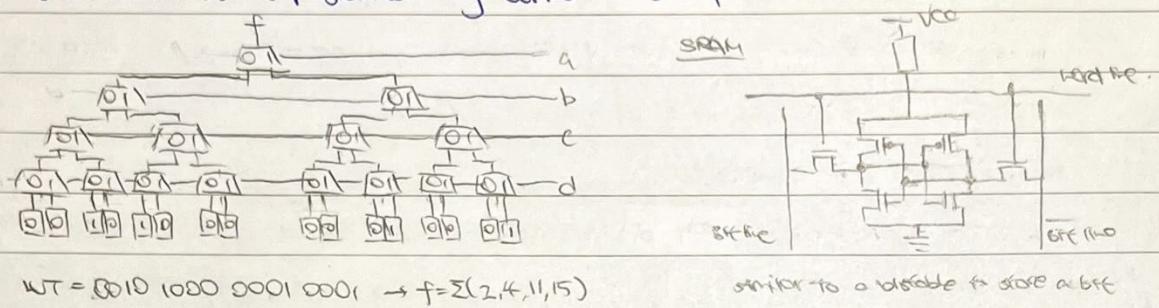
For Personal Use Only -bkwk2

Field programmable Gate Arrays (FPGA)

- FPGAs consist of LABs (based on logic elements), memory blocks, DSP blocks, PLLs and transceivers.
- Each LAB is made of logic elements (LE). A LE mainly consists of a LUT, carry chain and register.

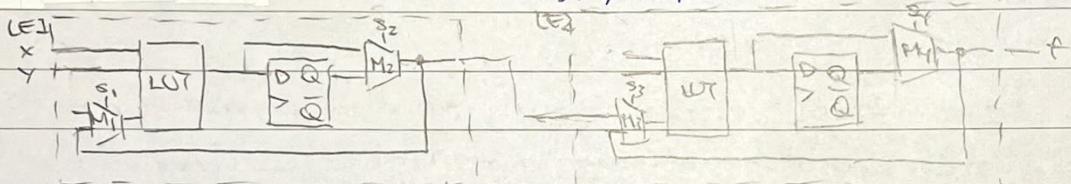


- A look-up table (LUT) replaces the product term array and is using memory as a combinational circuit.
- A LUT is formed using MUXes (where ext. ips are only used as select ips) and its contents (i.e. functionality) are programmed using EEPROM or SRAM.



- SRAM is volatile - req. power to maintain the stored information

- LEs can be used to implement sequential circuits. Typically we have:



$$S_1 S_2 S_3 S_4 = 1100 \rightarrow \text{LE1 as sequential circuit, LE2 as combinational circuit}$$

- LE2 LUT has ext. ip x,y → Mealy machine ; LE2 LUT no ext. ip x,y → Moore machine
- In a FPGA, the LABs are arranged in an array, connected w/ interconnect
 - ↳ local interconnect : connects btwn LEs within a LAB / direct connections btwn adjacent tiles
 - ↳ row and col. interconnect : fixed length routing segments that span a no of LABs / entire device.

- The I/O elements in FPGAs control I/O features (i.e. vs. o/p vs bidirectional, i/o standards etc)
 - ↳ controls tri-state buffer.
 - A typical I/O element consists of o/p enable control, o/p path and i/p path

- FPGAs use PLLs to generate clocks and DLLs to lock the phase quickly. The clock control blocks enable/disable clocks and control the clock routing network (which span the entire device).
- Most FPGAs use SRAM to program PI and LUT → use ext. EEPROM, CPLD or CPU to program
- (ACTIVE: FPGA controls program sequence automatically ON ; POSSIVE: CPU controls program sequence).

- Advantages of FPGA :

- ↳ High performance (high density to create complex logic performance), integration of many functions
- ↳ fast programming, low cost (compared to ASIC for low quantities)

For Personal Use Only -bkwk2

Data storage, processing and control!

Data storage.

- The main types of memory are ROM and RAM.

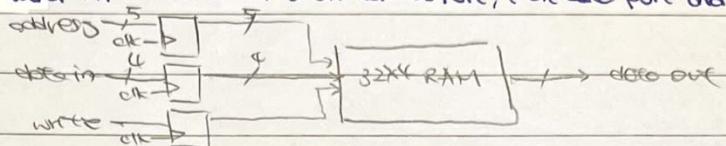
↳ Read-only memory (ROM): read only, not suitable as working space.

↳ Random access memory (RAM): read and write, much faster.

- Both ROM and RAM are random access — all addresses are accessible in an equal amount of time and can be selected in any order.

- FPGAs have dedicated memory blocks that can be constructed into ROM or sequential part RAM some memory blocks are memory LABs (MLABs) — LABs configured as memory.

↳ e.g.: 32 4-bit word RAM module w/ 5-bit address port, 4-bit data port and a write control input.

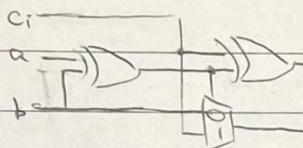


one address for read/write

one address for read
the address for write

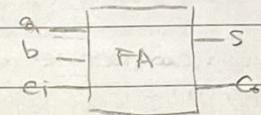
Arithmetic circuits.

- A full-adder (FA) circuit works as follows

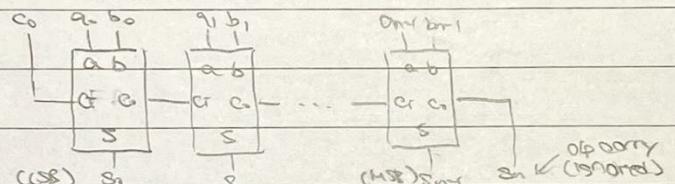


$$s = a \oplus b \oplus c_i$$

$$c_o = (\overline{a} \cdot \overline{b}) \cdot b + (a \cdot \overline{b}) \cdot c_i \\ = ab + a \cdot \overline{a} \cdot b \cdot c_i$$



- We can cascade n FAs to make an adder of n bit words. — ripple carry adder circuit



- We can invert one of the inputs and set $c_o=1$ to add the 2's complement → subtraction.

- We can make a comparator based on subtractors

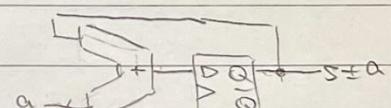
↳ $a=b$ when $a-b=0$ → check if it is zero by using n-bit NOR gate $a-1 \rightarrow D^n \rightarrow$

↳ $a < b$ when $a-b < 0$ → check MSB (the number has 1 as MSB) $b-1 \rightarrow D^n \rightarrow$

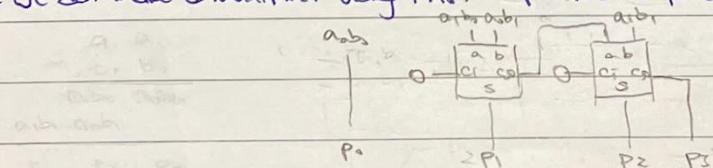
↳ $a > b$ when $a-b$ is not ≤ 0 or < 0 → pass =, $<$ or $>$ into NOR gate

- We can make an accumulator using an adder and a register

(counters are accumulators w/ $a=1$ always)



- We can make a multiplier using FAs. — $P = a \times b$ computed as an addition of summands

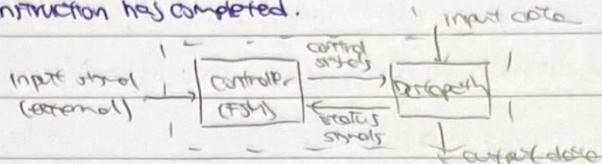


$$\begin{array}{r} a_1 \quad a_0 \\ \times \quad b_1 \quad b_0 \\ \hline a_1 b_1 \quad a_1 b_0 \\ a_0 b_1 \quad a_0 b_0 \\ \hline P_2 \quad P_1 \quad P_0 \end{array}$$

For Personal Use Only -bkwk2

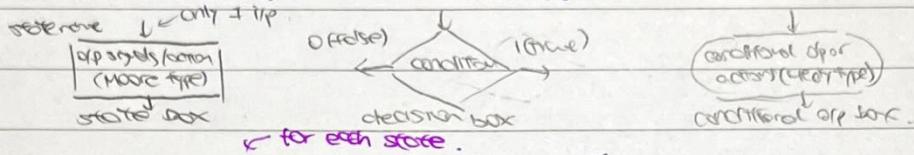
Digital systems

- A digital system consists of a control unit FCM (processor) that controls registers, MUXes and adder/subtractors. It executes operations specified in the form of instructions.
- The FCM goes through the instructions, asserting the control signals needed in successive clock cycles until the instruction has completed.



- The control circuit implements an algorithm — finite set of instructions which terminate in a finite time or a known end state. This has many representations — pseudocode, ASM charts etc.

- Algorithmic State Machine (ASM) charts contain three basic elements:



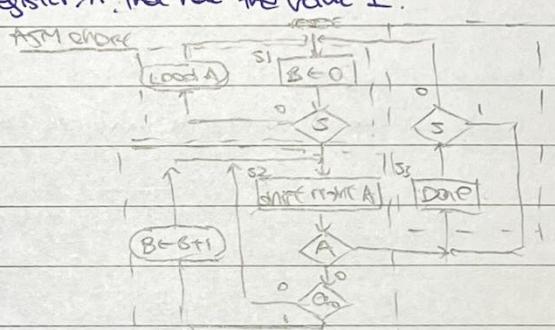
- In ASM charts, the entire block is considered as one unit — all operations within block occurring during single edge transition, (whereas in a conventional flowchart, one action follows another)
- The next state is evaluated during the same clock. System then enters the next state during transition of the next clock.

e.g.: we wish to count the no of bits in a register, A, that have the value 1.

Pseudocode

```

B=0;
while A<>0 do
    if A0=1 then
        B=B+1;
    end if;
    right-shift A;
end while;
    
```



For Personal Use Only -bkwk2

-e.g: Implement a simple synchronous circuit for the following traffic light controller.

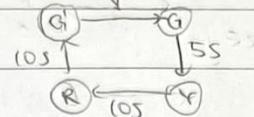
- Initial state vehicle GREEN

- After a pedestrian request is made, (i) change state vehicle YELLOW for 5 seconds,

(ii) change state vehicle RED for 10 seconds (iii) change state vehicle GREEN' for 10 seconds,

(iv) return to initial state.

① State diagram



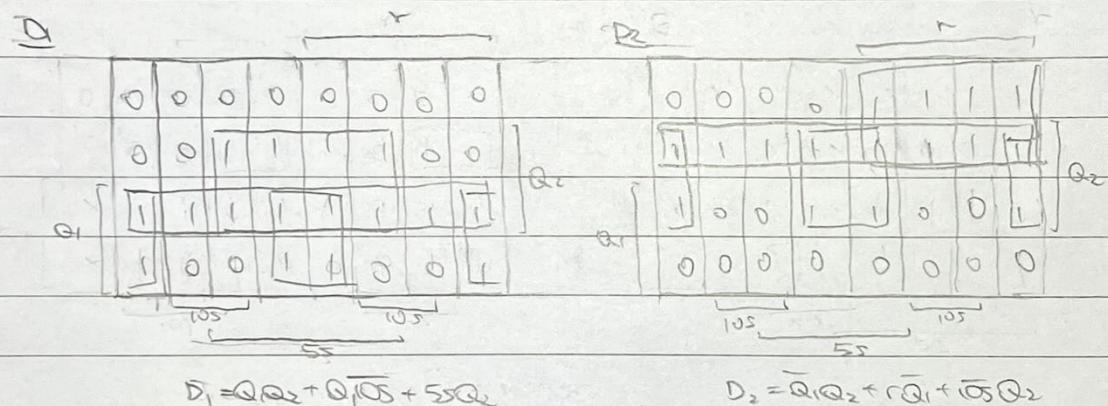
② State table + bisable allocation

| Present state | | Next state ($r, 5s, 10s$) | | | | | | | | Output | | doesn't depend | |
|---------------|------------|-----------------------------|-----|-----|-----|-----|-----|-----|-------------|--------|-----|----------------|------------------|
| State | Q_1, Q_2 | 000 | 001 | 011 | 010 | 110 | 111 | 001 | 100 | G | Y | R | on $r, 5s, 10s$ |
| G | 0 0 | | | | | | | Y | Y Y Y Y | 1 | 0 0 | 0 | → more realistic |
| Y | 0 1 | | | | | | | R | R R R R | 0 | 1 0 | | |
| R | 1 1 | | | | | | | G' | G' G' G' G' | 0 0 1 | | | |
| G' | 1 0 | | | | | | | G | G G G G | 1 0 0 | | | |

③ State transition table

| Present state | | Next state (Q_1, Q_2) | | | | | | | | Bisable inputs (D_1, D_2) | | Output | |
|---------------|------------|---------------------------|-----|-----|-----|-----|-----|-----|-----|-------------------------------|-----|--------|-----|
| State | Q_1, Q_2 | 000 | 001 | 011 | 010 | 110 | 111 | 001 | 100 | 000 | 001 | 010 | 011 |
| G | 0 0 | 00 | 00 | 00 | 00 | 01 | 01 | 01 | 00 | 00 | 00 | 01 | 01 |
| Y | 0 1 | 01 | 01 | 11 | 11 | 11 | 11 | 01 | 01 | 01 | 01 | 11 | 11 |
| R | 1 1 | 11 | 10 | 10 | 11 | 11 | 11 | 11 | 10 | 10 | 11 | 11 | 11 |
| G' | 1 0 | 10 | 00 | 00 | 10 | 10 | 00 | 10 | 10 | 00 | 00 | 10 | 10 |

④ Karnaugh maps



| | | |
|--|--|---|
| $\begin{array}{ c c }\hline Q_1 & Q_2 \\ \hline 1 & 1 \\ \hline 1 & 0 \\ \hline 0 & 0 \\ \hline \end{array}$ $G = \overline{Q}_2$ | $\begin{array}{ c c }\hline Q_1 & Q_2 \\ \hline 0 & 0 \\ \hline 1 & 0 \\ \hline 1 & 1 \\ \hline \end{array}$ $Y = \overline{Q}_1 Q_2$ | $\begin{array}{ c c }\hline Q_1 & Q_2 \\ \hline 0 & 0 \\ \hline 0 & 1 \\ \hline 1 & 1 \\ \hline \end{array}$ $R = Q_1 Q_2$ |
|--|--|---|

⑤ Circuit diagram

This can be implemented in many ways: logic gates, PLA, PAL, ROM, FPGA etc.

For Personal Use Only -bkwk2

VHDL

VHDL books

VHDL design write

- VHDL code for a high-level component has the following structure.

entity high_level_component is

 generic declaration

 port declaration

end high_level_component;

architecture

 component(s) declaration

 type declaration

 constant declaration

 signal declaration

begin

 component instantiation — genericmap, portmap

end behaviour;

- VHDL code for a low-level component has the following structure.

entity low_level_component is

 generic declaration

 port declaration

end low_level_component;

architecture

 type declaration

 constant declaration

 signal declaration

begin

 concurrent block

 sequential block (process)

end

- sometimes it may be useful use libraries / modules

library IEEE;

use ieee.std_logic_1164.all;

For Personal Use Only -bkwk2

① Generic declaration — parameterises the entity

↳ e.g.: generic (n: NATURAL := 4; K: INTEGER);

② Port declaration — describes i/p/s and o/p/s

↳ e.g.: port (x1, x2: IN BIT;

f: OUT BIT);

③ Type declaration — defines custom datatypes (useful for states)

↳ e.g.: type state-type is (s1, s2, s3);

④ Constant declaration — defines constants

↳ e.g.: constant k: INTEGER := 3;

⑤ Signal declaration — defines signals (connectors)

↳ e.g.: signal y: state-type;

⑥ Component declaration — includes low-level component in high-level component

↳ e.g.: component counter is

generic (n: NATURAL := 4);

port (clock: IN STD_LOGIC; count: OUT STD_LOGIC_VECTOR(n downto 0));

end component;

⑦ Component instantiation — configures the low level component

↳ e.g.: slow_clock: counter

generic map (n => 27);

port map (clock => CLOCK_50, count => BCD0);

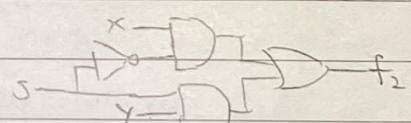
concurrent assignment statements

-the order in which they appear in VHDL code does not affect the meaning of the code

(represent implied processes that execute in parallel)

(i) Simple signal assignment

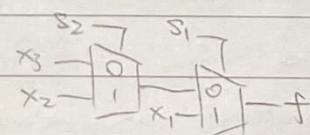
↳ e.g.: f2 <= (NOT s AND x) OR (s AND y);



(ii) Conditional signal assignment (2-to-1-MUX)

↳ e.g.: f <= x1 when s1 = '1' else

x2 when s1 = '0' else x3;



(iii) Selected signal assignment (general MUX)

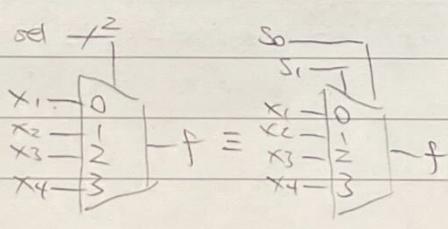
↳ e.g.: with sel select

f <= x1 when "00",

x2 when "01",

x3 when "10",

x4 when OTHERS;



For Personal Use Only -bkwk2

sequential assignment statements

- The ordering of the statements may affect the meaning of the code
(placed inside a process statement — explicit processes).

- A process has the following structure :

process_name : process (sensitivity_list)

 type declaration

 constant declaration

 variable declaration

 begin

 sequential statement

 end process

① Variable declaration — defining variables (local variables)

e.g.: variable count : INTEGER := '0';

② Sequential statement — increase behaviour and express order

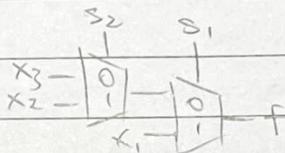
(i) if-then statements

e.g.: if s1='1' then f <= x1;

 elsif s2='1' then f <= x2;

 else f <= x3;

 end if;



(ii) case statements (general mux)

e.g.: case sel is

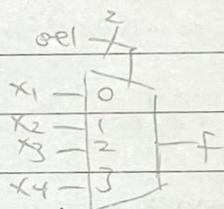
 when "00" => f <= x1;

 when "01" => f <= x2;

 when "10" => f <= x3;

 when OTHERS => f <= x4;

end case;



signal and variable assignment

- Signals are global and can be assigned values using the \leftarrow operator

- Variables are local (to processes) and can be assigned values using the $:=$ operator

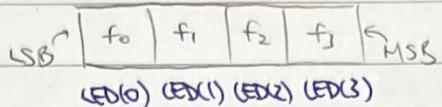
For Personal Use Only -bkwk2

std_logic_vector data type .

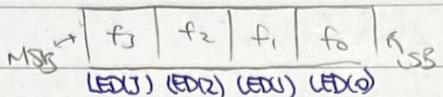
The std_logic_vector data type is part of the ieee.std_logic_1164 module.

- A std_logic_vector can be defined using TO or COUNTTO.

e.g.: LED : OUT std::logic_vector<0 TO 3>



LED = OUT sed-logic-vector(3 DOWNTO 0)



- Using **DOWN TO** is generally more natural, but we use **TO** by convention in certain applications (e.g. 7-segment display)

- we can slice a std::logic_vector in the following ways:

↳ All bits: $LGD \Leftarrow "000"$

↳ single bit LED(0) <= '0';

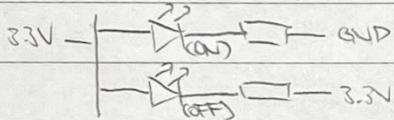
↳ slicing : `LED(3 DOWNTO 2) <= "10";` or `LED(2 TO 3) <= "01";`

7-segment display

$$\begin{array}{r} 6 \\ \times 5 \\ \hline 30 \end{array}$$

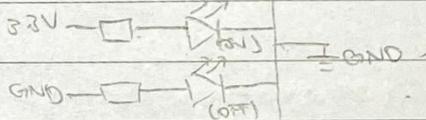
- 7-segment displays display decimal numbers. (LED segments are connected to common pins (CP))

① Common anode (+) ← used in
382 lab.



set 'd' to turn on LED

(2) Common cathode (-)



set '1' to turn on LED

In general, we can consider a controlled inverter \rightarrow change s as required.



$$\text{Common grade } \xi = 1, \quad \alpha = \frac{1}{\alpha'}$$

common cofactor $S=0$, $a=a'$

- We can implement a 7-segment decoder by considering the truth table \rightarrow K-map

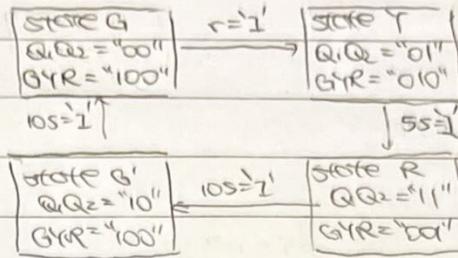
→ logic gate / PLA / PAL / ROM / LUT implementation

* The truth table for common anode and common cathode are inverses of each other.

For Personal Use Only -bkwk2

VHDL for sequential circuits.

- When designing a Moore machine, we can separate the FSM and its I/Os into two processes.
- e.g., consider the following state diagram



```

entity tlc is
  port(clk : IN STD_LOGIC;
        r, 5S, 10S : IN STD_LOGIC;
        G, Y, R : OUT STD_LOGIC);
end tlc;
  
```

architecture behaviour of tlc is

```

type state_type is (G, Y, R, G');
signal state : state_type;
begin
  FSM : process(clk)
    begin
      if rising-edge(clk) then
        case state is
          when G => if r='1' then state <= Y;
          else state <= G; end if;
          when Y => if 5S='1' then state <= R;
          else state <= Y; end if;
          when R => if 10S='1' then state <= G';
          else state <= R; end if;
          when G' => if 10S='1' then state <= G;
          else state <= G'; end if;
        end case;
      end if;
    end process;
  
```

alternatively use "wait until rising-edge(clk);"
or "if (clk'event AND clk='1') then"

For Personal Use Only -bkwk2

outputs : process (state)

begin

case state is

when G OR G' \Rightarrow Q = '1';

when Y \Rightarrow Y = '1';

when OTHERS \Rightarrow R = '1';

end case;

end process

end behaviour;