1P2J Digital Electronics 2 – Solutions

Mark Cannon Trinity Term 2026

1. Multiplexers

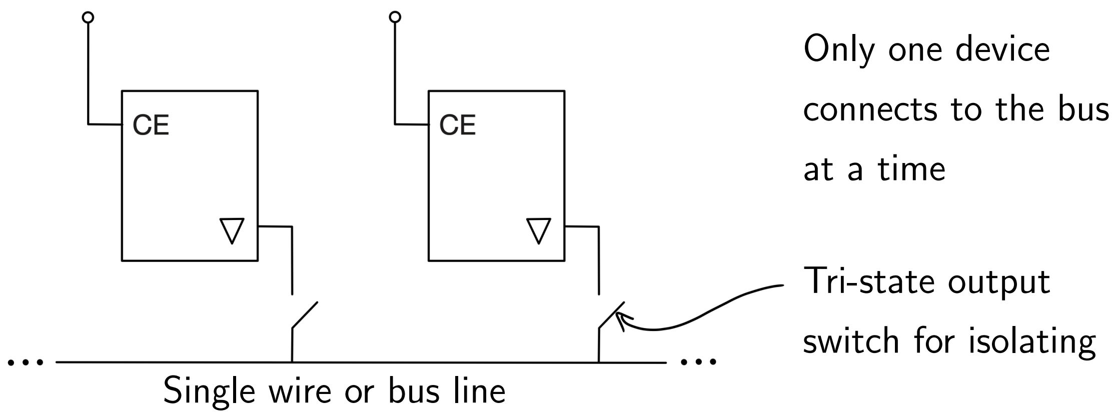
(a).

$\overline{\text{EN}}$	G_0	G_1	0	1	2	3	D_{out}
1	×	×	×	×	×	×	0
0	0	0	0	×	×	×	0
0	0	0	1	×	×	×	1
0	0	1	×	0	×	×	0
0	0	1	×	1	×	×	1
0	1	0	×	×	0	×	0
0	1	0	×	×	1	×	1
0	1	1	×	×	×	0	0
0	1	1	×	×	×	1	1

(b). Fan out is the number of logic gate inputs that can be connected to an output before the logic voltage levels become compromised. It is dependent on the type of logic family (e.g. TTL, CMOS, I2L, ECL etc.).

(c). Tri-state has 3 values: 0, 1 or high impedance.

The high impedance state effectively disconnects the output from the rest of the circuit, allowing data or address buses to work independently.



2. ROMs

(a). (i). EPROM – to allow modifications to software.

(ii). PLA – efficient use of number of gates;

(iii). EPROM – BIOS needs updating at times!

(iv). ROM – programmed at manufacture.

(b). A ROM fully decodes all input lines. This would require gates with very large numbers of inputs (see Hill & Peterson, p140 for an example).

3. PLAs

(a). A ROM, with a large number of inputs, must use a large number of gates to fully decode the input address. This is inefficient use of silicon chip area and results in propagation delays. A PLA, however, can be programmed to decode only prime implicants to obtain a more efficient realisation.

(b). Consider the sum $A+B=S$, where $A = A_n A_{n-1} \ldots A_0$, $B = B_n B_{n-1} \ldots B_0$, and $S = S_n S_{n-1} \ldots S_0$

The sum can be written as:

$$\begin{array}{rl} & A_n A_{n-1} \cdots A_0 \\ + & B_n B_{n-1} \cdots B_0 \\ \hline & S_n S_{n-1} \ldots S_0 \end{array} \quad \text{with carry out } C_n$$

Here $S_n = A_n + B_n + C_{n-1}$ and

(i) $A_n = B_n = 1 \;\Rightarrow\; C_n = 1$ $\qquad S_n = \begin{cases} 0 \text{ if } C_{n-1} = 0 \\ 1 \text{ if } C_{n-1} = 1 \end{cases}$

(ii) $A_n \neq B_n \;\Rightarrow\; C_n = \begin{cases} 0 \text{ if } C_{n-1} = 0 \\ 1 \text{ if } C_{n-1} = 1 \end{cases}$ $\qquad S_n = \begin{cases} 1 \text{ if } C_{n-1} = 0 \\ 0 \text{ if } C_{n-1} = 1 \end{cases}$

(iii) $A_n = B_n = 0 \;\Rightarrow\; C_n = 0$ $\qquad S_n = \begin{cases} 0 \text{ if } C_{n-1} = 0 \\ 1 \text{ if } C_{n-1} = 1 \end{cases}$

Truth table:

A_n	B_n	C_{n-1}	S_n	C_n	
0	0	0	0	0	(iii)
0	0	1	1	0	(iii)
0	1	0	1	0	(ii)
0	1	1	0	1	(ii)
1	0	0	1	0	(ii)
1	0	1	0	1	(ii)
1	1	0	0	1	(i)
1	1	1	1	1	(i)

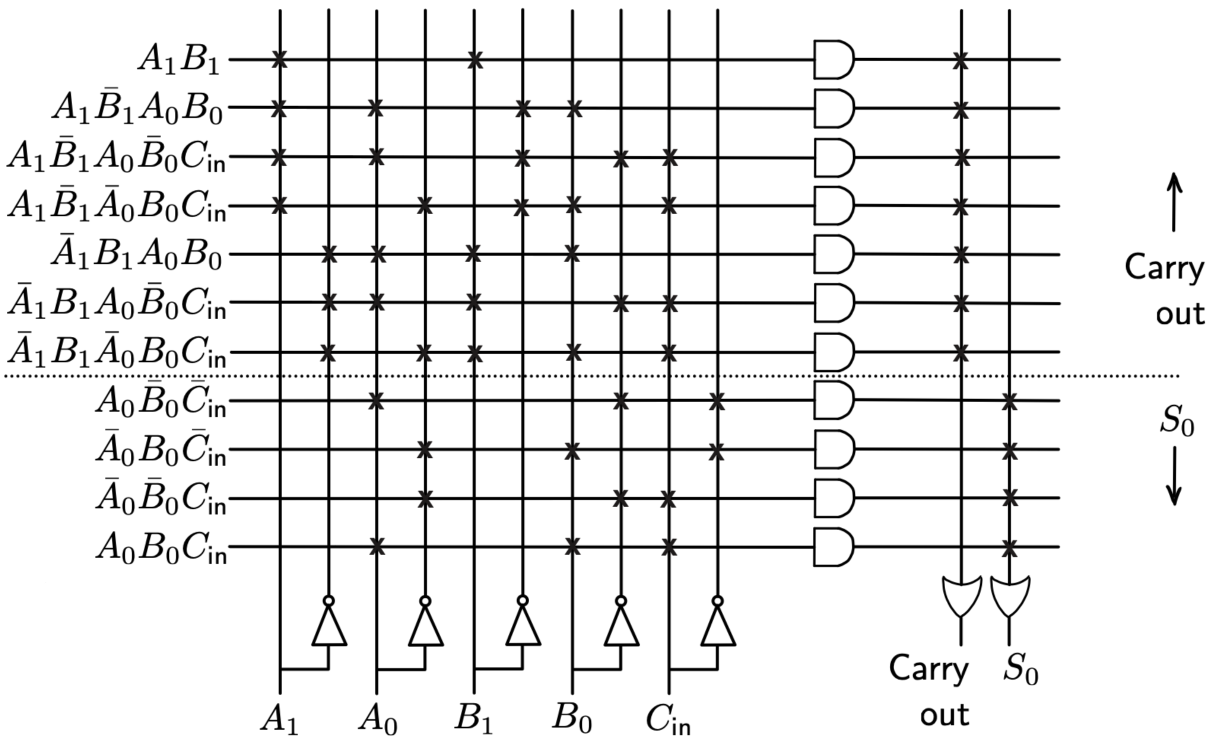
$$\Rightarrow \; C_n = A_n.B_n + (A_n.\bar{B}_n + \bar{A}_n.B_n).C_{n-1}$$
$$S_n = \big(A_n.\bar{B}_n + \bar{A}_n.B_n\big).\bar{C}_{n-1} + (\bar{A}_n.\bar{B}_n + A_n.B_n).C_{n-1}$$

(C_n can be simplified to $C_n = A_n.B_n + (A_n \oplus B_n).C_{n-1}$)

(c). The expressions for C_n and S_n derived in (b) give

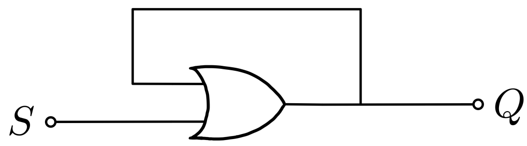
$$\begin{aligned}
\text{Carry out} = C_1 &= A_1.B_1 + (A_1\bar{B}_1 + \bar{A}_1.B_1).C_0 \\
&= A_1.B_1 + (A_1.\bar{B}_1 + \bar{A}_1.B_1).\big[A_0.B_0 + (A_0.\bar{B}_0 + \bar{A}_0.B_0).C_{\text{in}}\big] \\
&= A_1.B_1 + A_1.\bar{B}_1.A_0.B_0 + A_1.\bar{B}_1.A_0.\bar{B}_0.C_{\text{in}} + A_1.\bar{B}_1.\bar{A}_0.B_0.C_{\text{in}} \\
&\quad + \bar{A}_1.B_1.A_0.B_0 + \bar{A}_1.B_1.A_0.\bar{B}_0.C_{\text{in}} + \bar{A}_1.B_1.\bar{A}_0.B_0.C_{\text{in}} \\
S_0 &= (A_0.\bar{B}_0 + \bar{A}_0.B_0).\bar{C}_{\text{in}} + (\bar{A}_0.B_0 + A_0.B_0).C_{\text{in}} \\
&= A_0.\bar{B}_0.\bar{C}_{\text{in}} + \bar{A}_0.B_0.\bar{C}_{\text{in}} + \bar{A}_0.\bar{B}_0.C_{\text{in}} + A_0.B_0.C_{\text{in}}
\end{aligned}$$

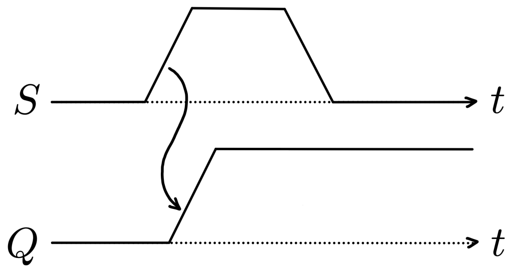
Therefore 7 terms for Carry out, 4 terms for S_0, and 11 terms in total.



4. State and memory

(a).





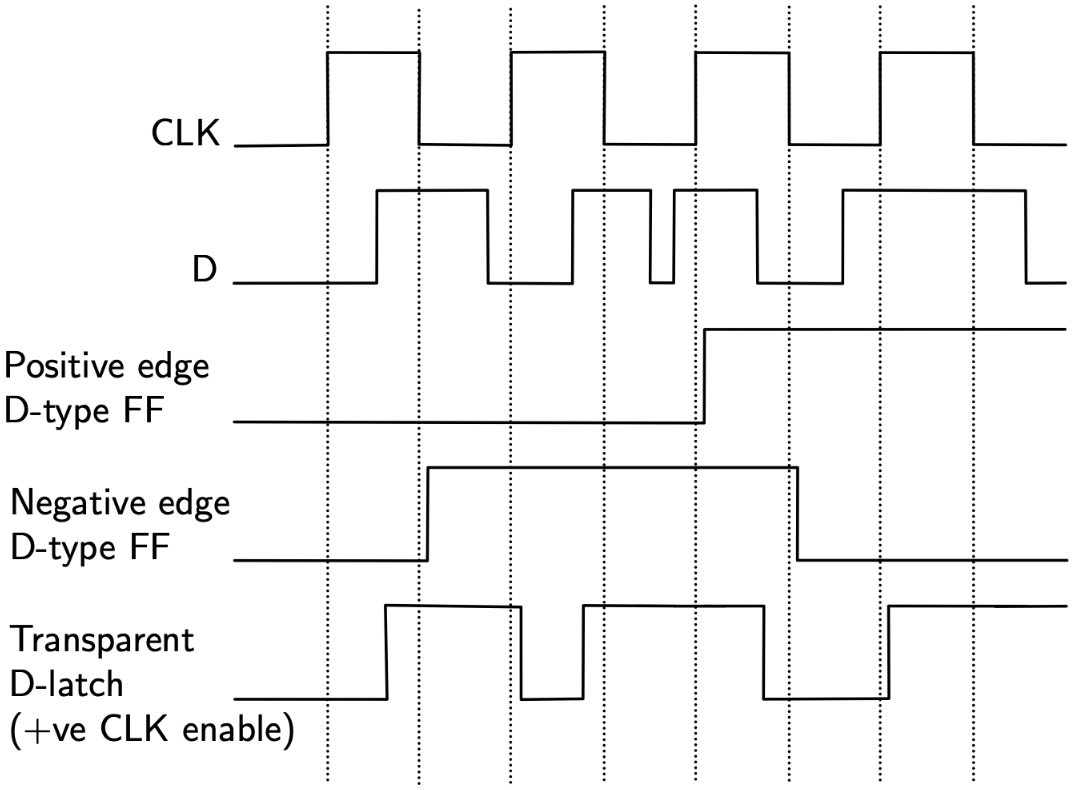
Q doesn't go back to 0, so it has recorded the event $S : 0 \to 1$, and it has memory.

But we need a way to reset the latch in order to make it useful for storing data.

(b). With n flip-flops we can represent 2^n states.

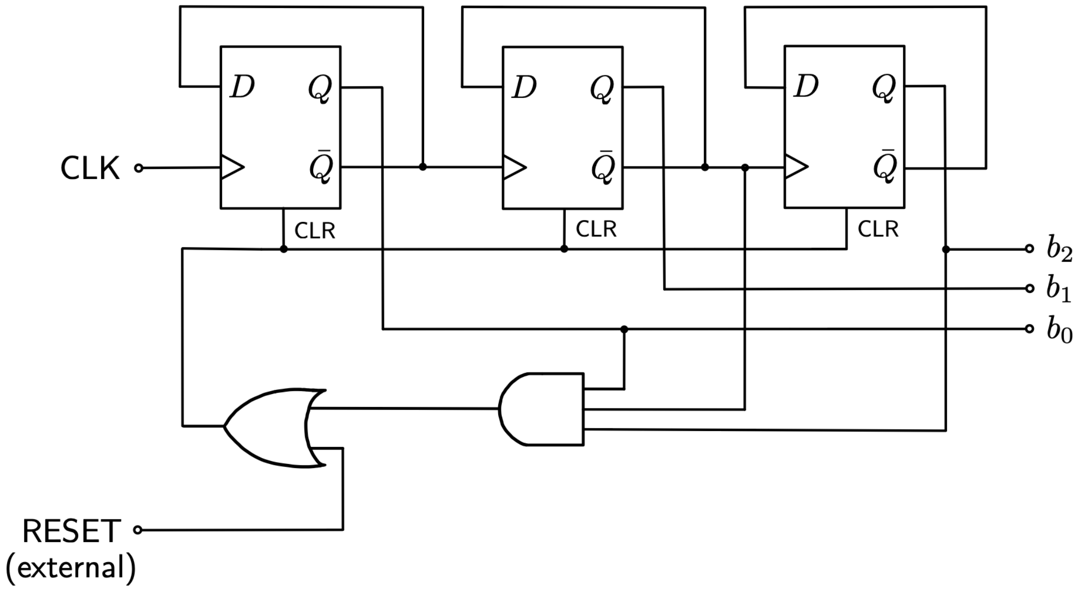
5. Timing diagrams for memory elements

Timing diagram:



6. Asynchronous counters

(a). Modulo-5 counter must reset to 000 when count reaches 101 ($= 5$)

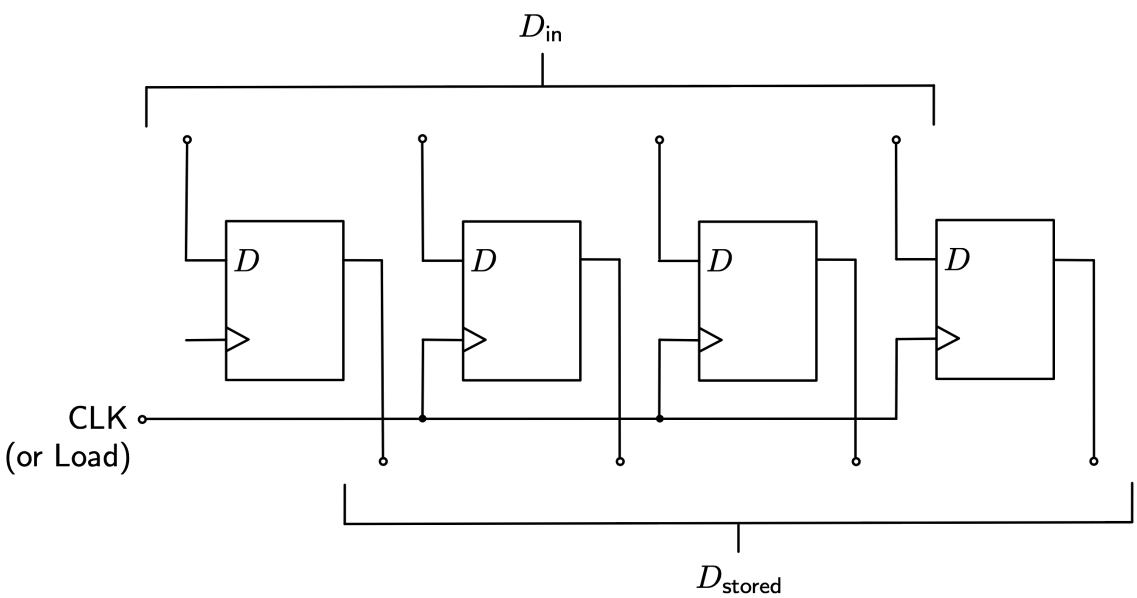


Note that asynchronous (external) RESET leads to a glitch in the timing diagram, as the count value 101 must exist for an instant before the flip-flops are cleared back to 000. Completely synchronous designs are better.

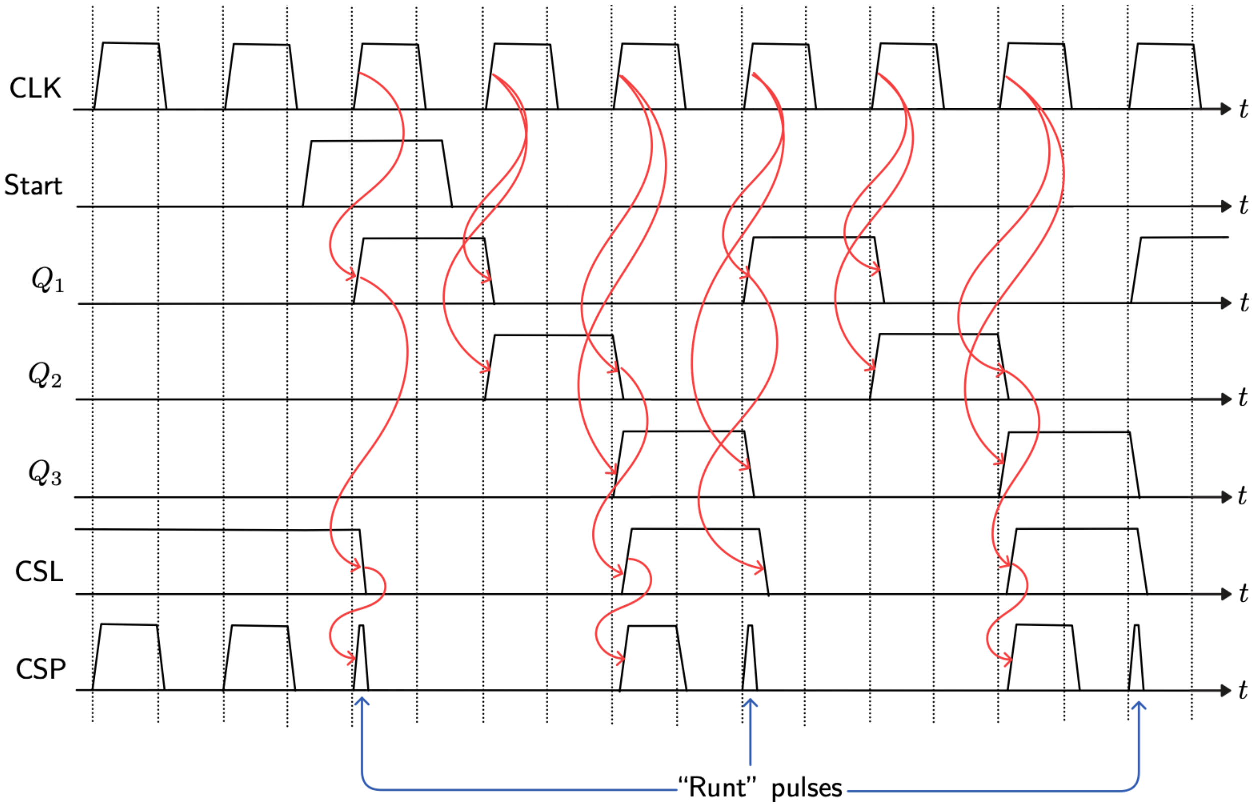
(b). The counter speed is determined by the propagation delay of the 3 FFs combined (like a ripple effect).

7. Synchronous logic, clocks, flip-flops

(a). 4-bit data register using D-type flip-flops:



(b). Timing diagram:



8. Synchronous counters

(a). Steering table for D-type:

Transition	D
$0 \to 0$	0
$0 \to 1$	1
$1 \to 1$	1
$1 \to 0$	0

State transition table:

b_2^n	b_1^n	b_0^n	$b_2^n \to b_2^{n+1}$	$b_1^n \to b_1^{n+1}$	$b_0^n \to b_0^{n+1}$
0	0	0	$0 \to 0$	$0 \to 0$	$0 \to 1$
0	0	1	$0 \to 0$	$0 \to 1$	$1 \to 0$
0	1	0	$0 \to 0$	$1 \to 1$	$0 \to 1$
0	1	1	$0 \to 1$	$1 \to 0$	$1 \to 0$
1	0	0	$1 \to 1$	$0 \to 0$	$0 \to 1$
1	0	1	$1 \to 0$	$0 \to 0$	$1 \to 0$

Karnaugh maps for $D_i = b_i^n$ and $D_i' = b_i^{n+1}$, $i = 0, 1, 2$:

$D_2 \backslash D_1D_0$	00	01	11	10
0	0	0	1	0
1	1	0	X	X

D_2'

$D_2' = D_1.D_0 + D_2.\bar{D}_0$

$D_2 \backslash D_1D_0$	00	01	11	10
0	0	0	1	0
1	1	0	X	X

D_1'

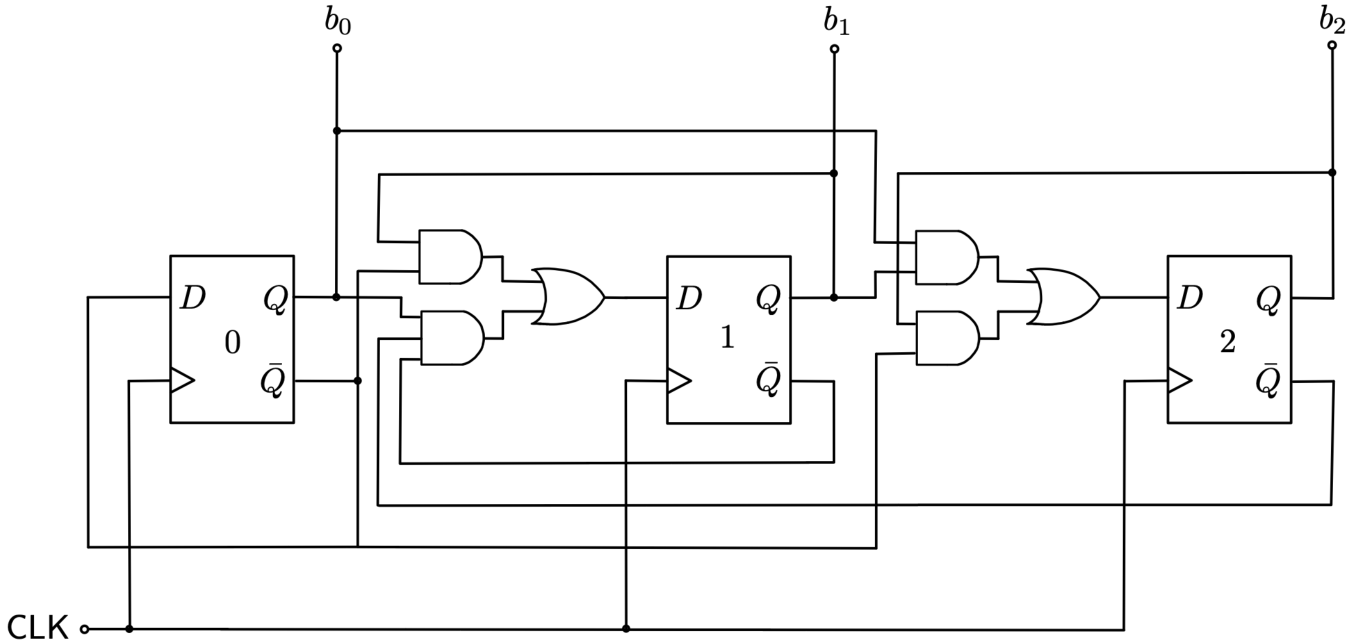
$D_1' = D_1.\bar{D}_0 + \bar{D}_2.\bar{D}_1.D_0$

$D_2 \backslash D_1D_0$	00	01	11	10
0	0	0	1	0
1	1	0	X	X

D_0'

$D_0' = \bar{D}_0$

Circuit diagram:



(b). The two don't care states are 110 $(= 6)$ and 111 $(= 7)$.

By examining the circuit above in both states we see that:

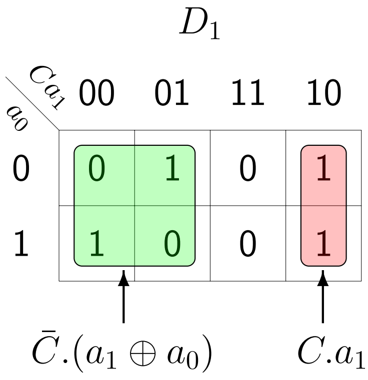
- State 110 is followed by state 111 (which is also unwanted).
- State 111 is followed by state 100 (which is valid). The correct sequence would then continue.

It's better practice to design the circuit to revert to state 000 if states 110 and 111 occur.

9. Simple state machines

(a). Sequence A:

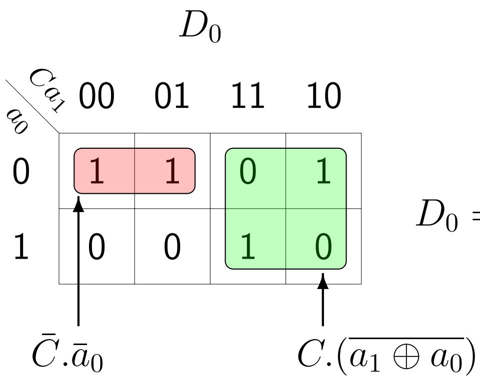
C	a_1	a_0
0	0	0
0	0	1
0	1	0
0	1	1



$$D_1 = C.a_1 + a_1 \oplus a_0$$

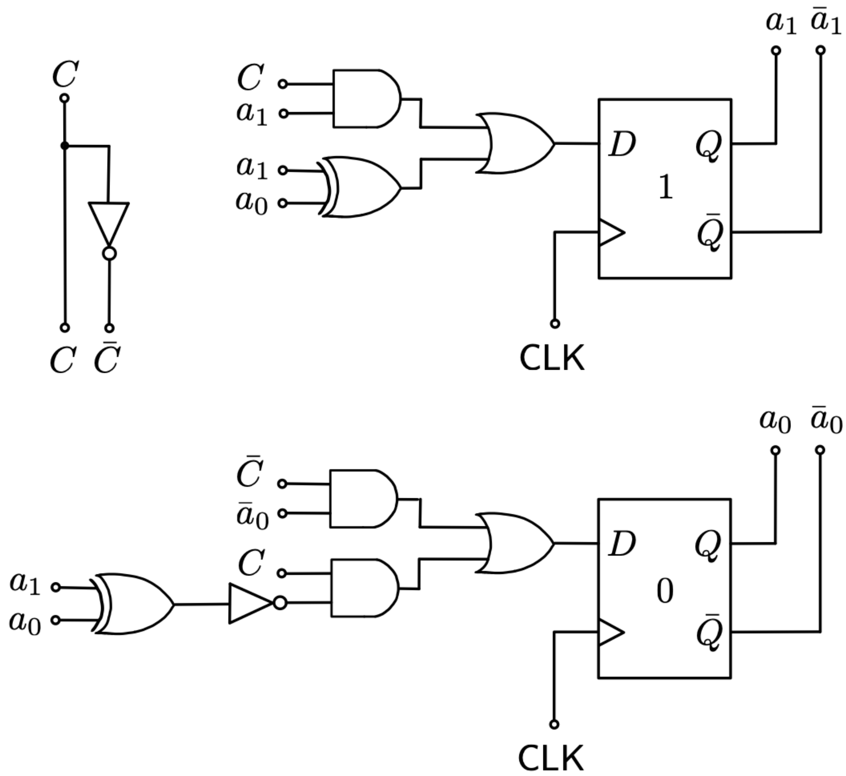
Sequence B:

C	a_1	a_0
1	0	0
1	1	1
1	0	1
1	1	0

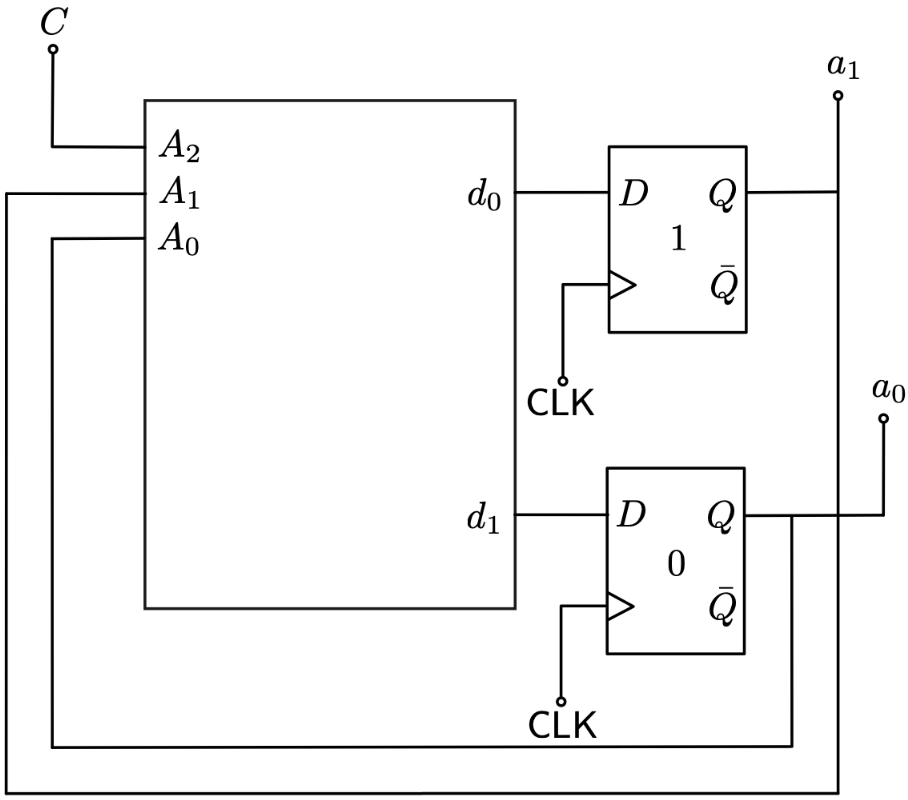


$$D_0 = \bar{C}.\bar{a}_0 + C.(\overline{a_1 \oplus a_0})$$

Circuit diagram:



(b).



(c).

address			contents	
A_2	A_1	A_0	d_1	d_0
0	0	0	0	1
0	0	1	1	0
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	0	1

10. Analogue to digital converter (ADC)

(a). The successive-approximation converter approximates the analogue input signal with a binary code. This binary code is successively revised by changing each bit in the code until the best approximation is achieved.

At each step in this process, the current estimate of the binary value corresponding to the analogue input signal is saved in the successive approximation register. The contents of this register are converted to an analogue signal by a DAC, and a comparator is used to determine whether the approximation is larger or smaller than the input signal.

Each bit is tested in decreasing order of value. After the least significant bit has been tested, the conversion is complete and the output register contains the binary code.

(b). 2 samples in 1 period $\Rightarrow$ one sample every $50\,\text{ms}$.

8-bit resolution means there are 8 bits per sample, 40 clock cycles are required per bit, so $8 \times 40 = 320$ clock cycles per sample.

Worst case is a sinusoidal signal at $10\,\text{kHz}$: $V(t) = \frac{1}{2}(1 + \sin\omega t)V_0$, with $\omega = 2\pi \times 10^4\,\text{rad s}^{-1}$ and $V_0 = 1\,\text{V}$ (full scale).

Then the maximum slew rate is $\dfrac{\mathrm{d}V}{\mathrm{d}t} = \frac{1}{2}\omega V_0 = \pi \times 10^4\,\text{Vs}^{-1}$.

We need the change in voltage during the conversion period to be no greater than half the voltage equivalent of the LSB, hence

$$\Delta V = t_c\frac{\mathrm{d}V}{\mathrm{d}t} \leq \frac{V_{\mathsf{LSB}}}{2} = 0.5\frac{1\,\text{V}}{256\,\text{bits}} \approx 1.95\,\text{mV}$$

Therefore the conversion period must satisfy

$$t_c \leq \frac{1.95 \times 10^{-3}}{\pi \times 10^4} = 62\,\text{ns}$$

This is equivalent to 320 clock cycles, so the clock period is $62\,\text{ns}/320 = 0.194\,\text{ns}$, which gives the minimum clock frequency as $5.16\,\text{GHz}$.

(c). The maximum frequency of a full amplitude sinusoidal signal, $\frac{1}{2}V_{\mathsf{REF}}\sin 2\pi f t$, is limited by the maximum permissible change in voltage during the conversion time: $f \leq f_{\max} = 1/(\pi 2^{n+1} t_c)$.

Using a sample and hold means that the conversion time t_c is replaced by the time, t_s, that's required to sample the signal, which is considerably shorter. Therefore higher frequencies can be sampled.

(d). The digitisation time for a flash ADC is very short, hence a sample and hold is not required for fast operation.

11. Digital to analogue converter (DAC)

(a). At the right hand end of the system there are two $2R$ resistors acting in parallel which combine to form an effective resistance R. This effective resistance then appears in series with another resistance R to form a resistance of $2R$. However, this effective resistance of $2R$ is in parallel with another resistance $2R$. Thus, at each stage of the analysis of the ladder network, all elements to the right of a particular node are equivalent to a resistance of $2R$.

An advantage of this architecture is that the inverting input of the op-amp is a virtual earth and hence one end of each the $2R$ resistors is always connected to 'earth'. This means that the current flowing through each branch of the ladder network is independent of the switch conditions and hence the digital input. This has the effect of ensuring that currents flow through each $2R$ resistor/switch path with ratio $8:4:2:1$ from left to right.

The input bits are represented by the positions of the switches (connected to the earth for 0 and to the op-amp for 1). The summing op-amp amplifier circuit in effect adds together the currents from each path, so that the DAC output voltage is determined by a current that is proportional to the weighted sum of the input bits.

(b). - For every stage, the input resistance is $2R$.

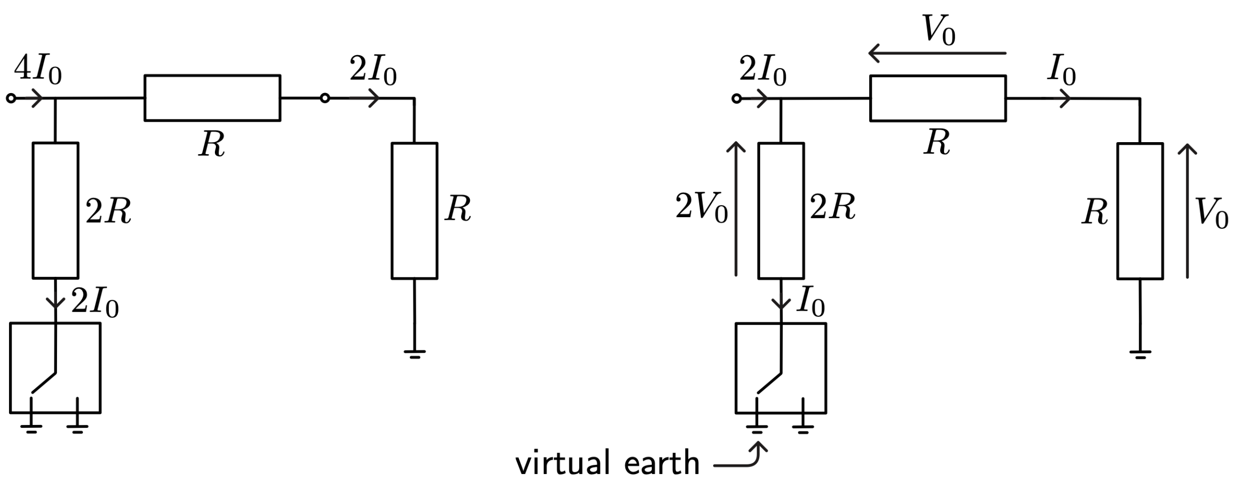
- The current into each stage is $2\times$ the current through each arm.

- The current through the lower arm passes through the switch.

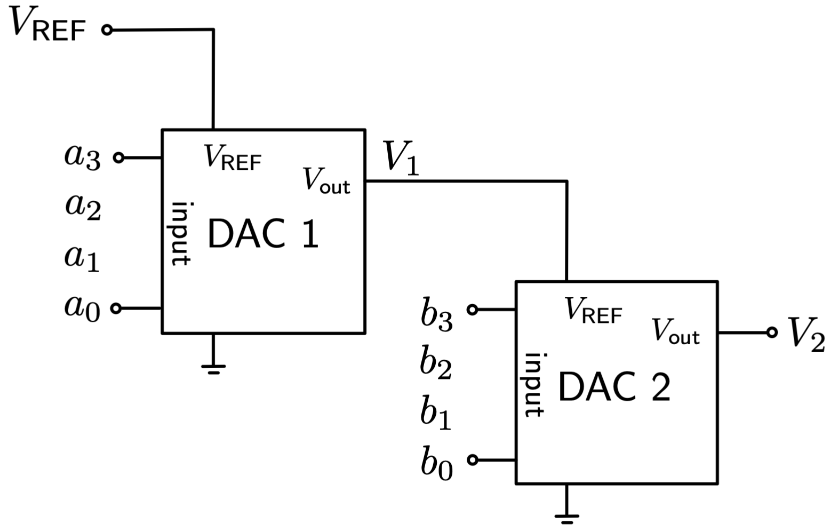
- The current through the upper arm becomes the input current to the next stage to the right.

- Hence, the current through a particular switch is twice the current passing through the next switch to the right.

- The currents through the switches are of the form $I_n = 2^n I_0$.



(c).



$$V_1 = -V_{\mathsf{REF}} \frac{8a_3 + 4a_2 + 2a_1 + a_0}{16}$$
$$V_2 = -V_1 \frac{8b_3 + 4b_2 + 2b_1 + b_0}{16}$$
$$= \frac{V_{\mathsf{REF}}}{256}(8a_3 + 4a_2 + 2a_1 + a_0)(8b_3 + 4b_2 + 2b_1 + b_0)$$

The output voltage V_2 is in the range:

$$0 \le V_2 \le V_{\mathsf{REF}} \frac{15^2}{256} \approx 0.88 V_{\mathsf{REF}}.$$

(d).