Fall 1998 CS150 - Problem Set 2: Solutions David Chinnery

Question 1.
Denote inputs to this bit A_i and B_i, outputs from this bit L_i $(A < B)$ and E_i $(A = B)$, and inputs from the higher bit L_{i+1} and E_{i+1}.

To determine L_i:

- If $L_{i+1} = 1$, then $L_i = 1$ (the most significant bits of A are are less than B).
- If $E_{i+1} = 1$, then $L_i = \overline{A} \cdot B$ ($L_{i+1} = 0$ as A and B bits were equal).
- Otherwise $B < A$ and $L_i = 0$.
- Hence, $L_i = L_{i+1} + E_{i+1} \cdot \overline{A} \cdot B$.

To determine E_i:

- If $E_{i+1} = 1$, then $E_i = A \cdot B + \overline{A} \cdot \overline{B}$.
- Otherwise, $A < B$ or $B < A$, and $E_i = 0$.
- Hence, $\begin{aligned} E_i &= E_{i+1}(A \cdot B + \overline{A} \cdot \overline{B}) \\ &= E_{i+1}\overline{(A \oplus B)} \end{aligned}$

To determine $\min_i(A, B)$:

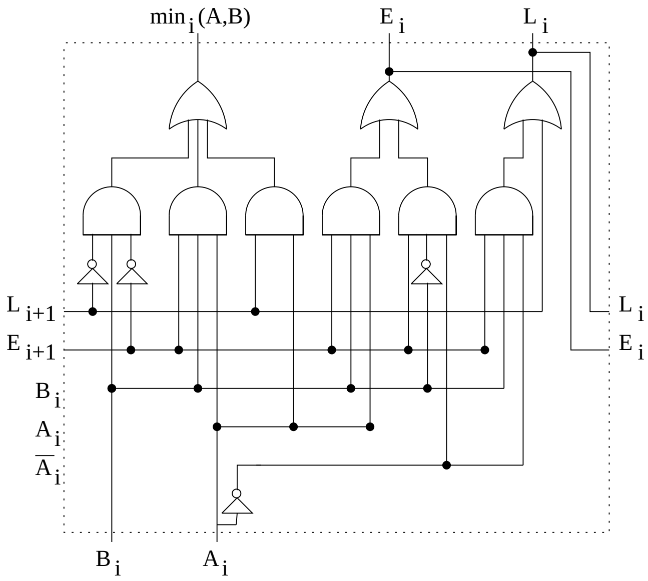
- If $L_{i+1} = 1$ then $\min_i(A, B) = A_i$.
- If $E_{i+1} = 1$ then $\min_i(A, B) = A_i \cdot B_i$. Either $A_i = B_i = 1$, or the minimum is zero.
- Otherwise $B < A$ and $\min_i(A, B) = B_i$.
- Hence, $\min_i(A, B) = L_{i+1} \cdot A_i + E_{i+1} \cdot A_i \cdot B_i + \overline{L}_{i+1} \cdot \overline{E}_{i+1} \cdot B$.

	L_{i+1}	E_{i+1}	L_i	E_i	$\min_i(A,B)$
$B < A$	0	0	0	0	B_i
$A = B$	0	1	$\overline{A}_i \cdot B_i$	$\overline{(A \oplus B)}$	$A_i \cdot B_i$
$A < B$	1	0	1	0	A_i
illegal	1	1			

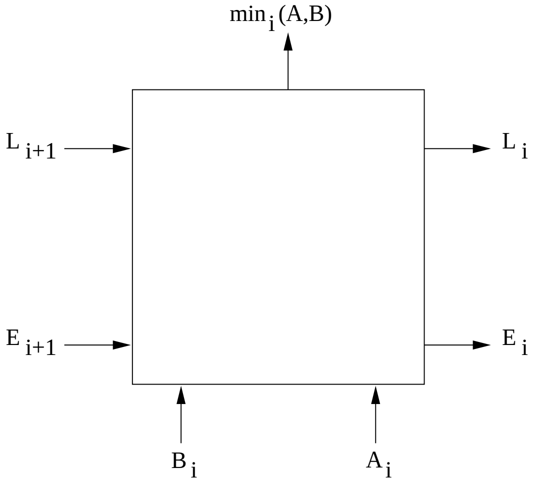
Table 1: Truth table for $\min_i$, L_i and E_i.

Shown below is a modular, combinatorial circuit which implements these functions and is easily cascaded.

In order for the combinatorial logic modules to correctly determine $\min(A, B)$, the most significant bit requires inputs of $E_8 = 1$ and $L_8 = 0$ (these are the inputs which would arrive from a higher significant bit to the module).



Modular logic diagram.



Block diagram of the inputs and outputs for the logic.

Figure 1: Logic and block diagrams for the modular circuit to compare A and B.

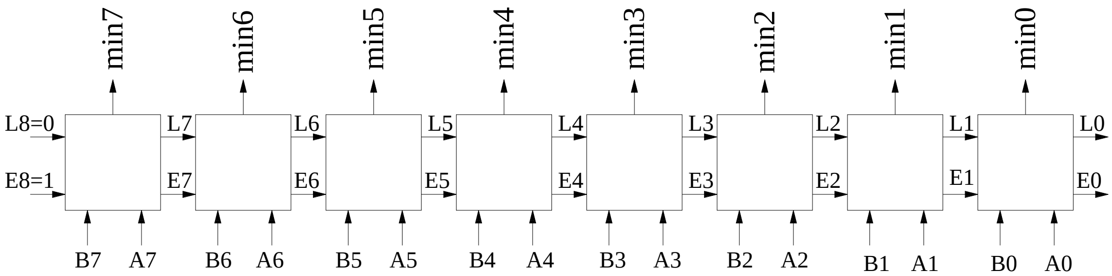


Figure 2: (b) Interconnection of 1 bit modules to implement 8 bit combinatorial logic, which computes the minimum of two 8 bit numbers A and B.

(c) To calculate the maximum delay, assuming unit delay for all gates and interters:

- Consider the most significant bit: E_8 and L_8 do not change, so the critical paths to consider are those for A_7 and B_7. To get the correct E_7 and L_7 outputs for the next stage: the longest paths for A_7 and B_7 are through an inverter, an AND gate, then an OR gate - taking **3 time units**.

- For lower significant bits, inputs A_i and B_i will be held constant until all the modules have finished - they have been stable for 3 units (or more) of time already. Whereas, the module cannot correctly compute E_i and L_i until E_{i+1} and L_{i+1} arrive - hence E_{i+1} and L_{i+1} are the critical signals to consider.

  To determine E_i and L_i, E_{i+1} and L_{i+1} signals must pass through one AND gate and one OR gate - a delay of 2 time units. There are six modules between module 7 and module 0, so $2 \times 6 =$ **12 time units**.

- Module 0 receives E_1 and L_1, what is the maximum delay to determine $\min_0(A, B)$? At worst, E_1 and L_1 have to propagate through an inverter, an AND gate, and an OR gate - a further **3 time units**.
- Thus, the worst case delay for calculating the 8 bit minimum is **18 time units**.

Question 2.
Note: it is assumed that S and R have the same values for $t < 0$ as at $t = 0$, otherwise Q_a and Q_b are unknown until $t = 1$ unit.

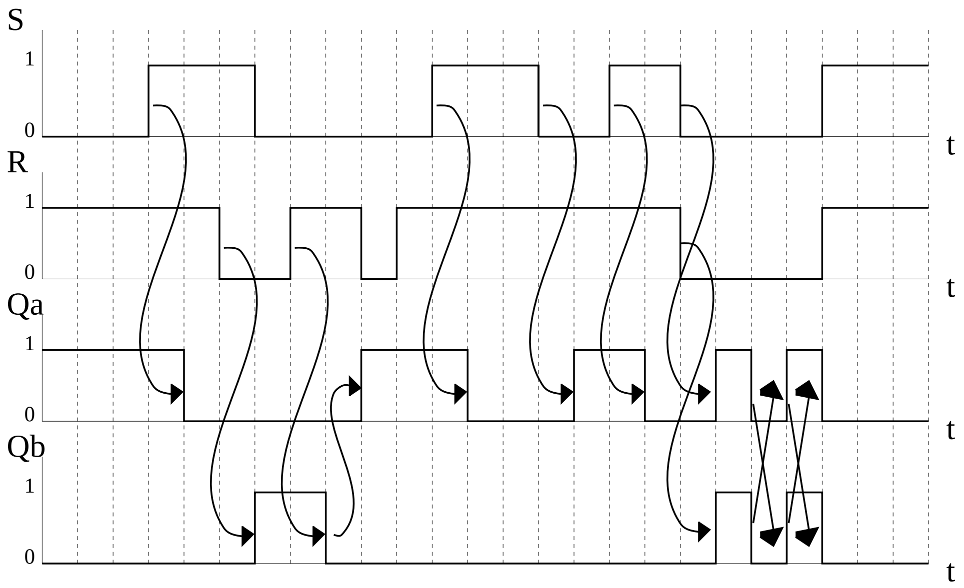


Figure 3: Timing diagram for the latch. Note oscillations occuring at the end of the waveform when $R = S = 0$, due to Q_a=Q_b.

Question 3.

Initial State	Q_1	Q_0	A	D_1	D_0	Next State
S_0	0	0	0	1	1	S_3
S_0	0	0	1	1	1	S_3
S_1	0	1	0	1	1	S_3
S_1	0	1	1	1	0	S_2
S_2	1	0	0	0	1	S_1
S_2	1	0	1	0	0	S_0
S_3	1	1	0	1	1	S_3
S_3	1	1	1	1	1	S_3

Table 2: State transition table, with $D_0 = \overline{A} + \overline{Q}_1 \oplus Q_0$ and $D_1 = \overline{Q}_1 + Q_0$.

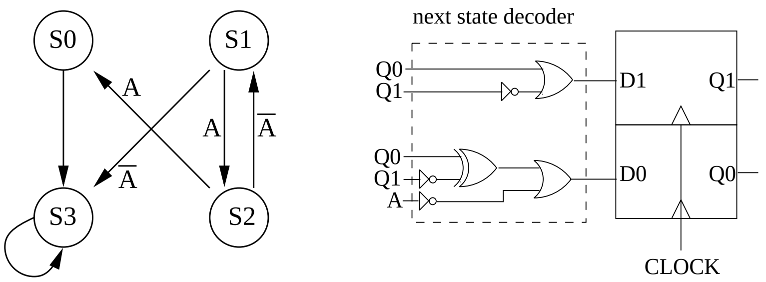


Figure 4: State diagram and logic for the finite state machine, tabulated in table 2.

Question 4.

Initial State	Q_2	Q_1	Q_0	D_2	D_1	D_0	Next State
S_0	0	0	0	0	0	1	S_1
S_1	0	0	1	0	1	0	S_2
S_2	0	1	0	1	0	0	S_4
S_3	0	1	1	1	0	1	S_5
S_4	1	0	0	1	1	0	S_6
S_5	1	0	1	1	1	1	S_7
S_6	1	1	0	0	1	1	S_3
S_7	1	1	1	0	0	0	S_0

Table 3: State transition table a 3 bit counter $Q_2Q_1Q_0$, counting from 0 to 7 repeatedly.

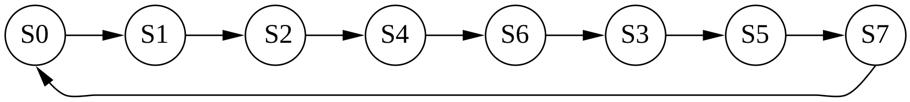


Figure 5: State diagram for the 3 bit counter, tabulated in table 3.

The logic implementation for the 3 bit counter, from table 3, is:

- $$\begin{aligned} D_2 &= \overline{Q}_2 Q_1 \overline{Q}_0 + \overline{Q}_2 Q_1 Q_0 + Q_2 \overline{Q}_1 \overline{Q}_0 + Q_2 \overline{Q}_1 Q_0 \\ &= \overline{Q}_2 Q_1 + Q_2 \overline{Q}_1 \\ &= Q_2 \oplus Q_1 \end{aligned}$$

- $$\begin{aligned} D_1 &= \overline{Q}_2 \overline{Q}_1 Q_0 + Q_2 \overline{Q}_1 \overline{Q}_0 + Q_2 \overline{Q}_1 Q_0 + Q_2 Q_1 \overline{Q}_0 \\ &= \overline{Q}_1 Q_0 + Q_2 \overline{Q}_0 \end{aligned}$$

- $$\begin{aligned} D_0 \quad & \overline{Q}_2 \overline{Q}_1 \overline{Q}_0 + \overline{Q}_2 Q_1 Q_0 + Q_2 \overline{Q}_1 Q_0 + Q_2 Q_1 \overline{Q}_0 \\ &= \overline{Q}_2 \oplus \overline{Q}_1 \oplus \overline{Q}_0 \end{aligned}$$

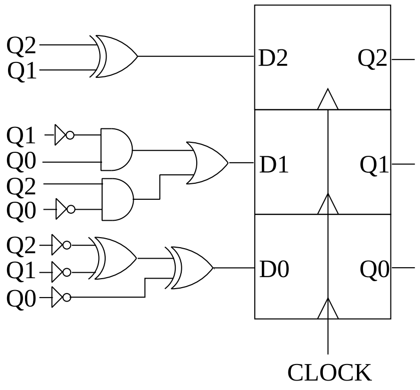


Figure 6: Schematic for the 3 bit counter.

Question 5.
The state table for a Mealey finite state machine which implements the minimum function on two bit streams A and B is shown in table 4. From the truth table, the logic required is:

$$\begin{aligned}
D_1 &= Q_1 + \overline{Q}_1\overline{Q}_0 A\overline{B} \\
&= Q_1 + \overline{Q}_0 A\overline{B} && \text{(either } Q_1 = 1 \text{ or } \overline{Q}_0 A\overline{B} \text{ with } Q_1 = 0\text{)} \\
D_0 &= Q_0 + \overline{Q}_1\overline{Q}_0 \overline{A}B \\
&= Q_0 + \overline{Q}_1 \overline{A}B && \text{(either } Q_0 = 1 \text{ or } \overline{Q}_1 \overline{A}B \text{ with } Q_0 = 0\text{)} \\
\min(A,B) &= \overline{Q}_1\overline{Q}_0 AB + Q_0 A + Q_1 B \\
&= \overline{Q}_1 AB + Q_0 A + Q_1 B && \text{(similar argument as above for } Q_0 \text{ A)}
\end{aligned}$$

This takes into account "don't cares" arising from state S_3 (namely $Q_1 = 1$ and $Q_0 = 1$ are mutually exclusive). The corresponding state diagram is shown in figure 7. Included is an external reset to reset the state to S_0.

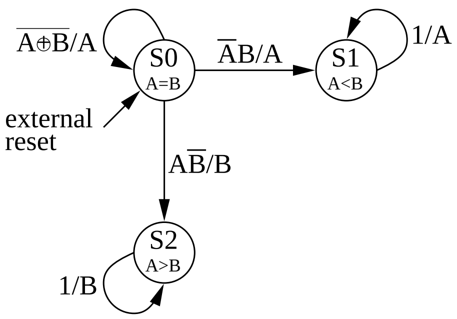


Figure 7: State diagram for the Mealey finite state machine to calculate $\min(A, B)$. The output is indicated after the /.

Initial State	Q_1	Q_0	A	B	$\min(A, B)$	D_1	D_0	Final State
S_0 equal	0	0	0	0	0	0	0	S_0
S_0	0	0	0	1	0	0	1	S_1
S_0	0	0	1	0	0	1	0	S_2
S_0	0	0	1	1	1	0	0	S_0
S_1 $A < B$	0	1	0	0	0	0	1	S_1
S_1	0	1	0	1	0	0	1	S_1
S_1	0	1	1	0	1	0	1	S_1
S_1	0	1	1	1	1	0	1	S_1
S_2 $B < A$	1	0	0	0	0	1	0	S_2
S_2	1	0	0	1	1	1	0	S_2
S_2	1	0	1	0	0	1	0	S_2
S_2	1	0	1	1	1	1	0	S_2

Table 4: State table for the Mealey finite state machine which calculates $\min(A, B)$ (note that state S_3 is illegal).

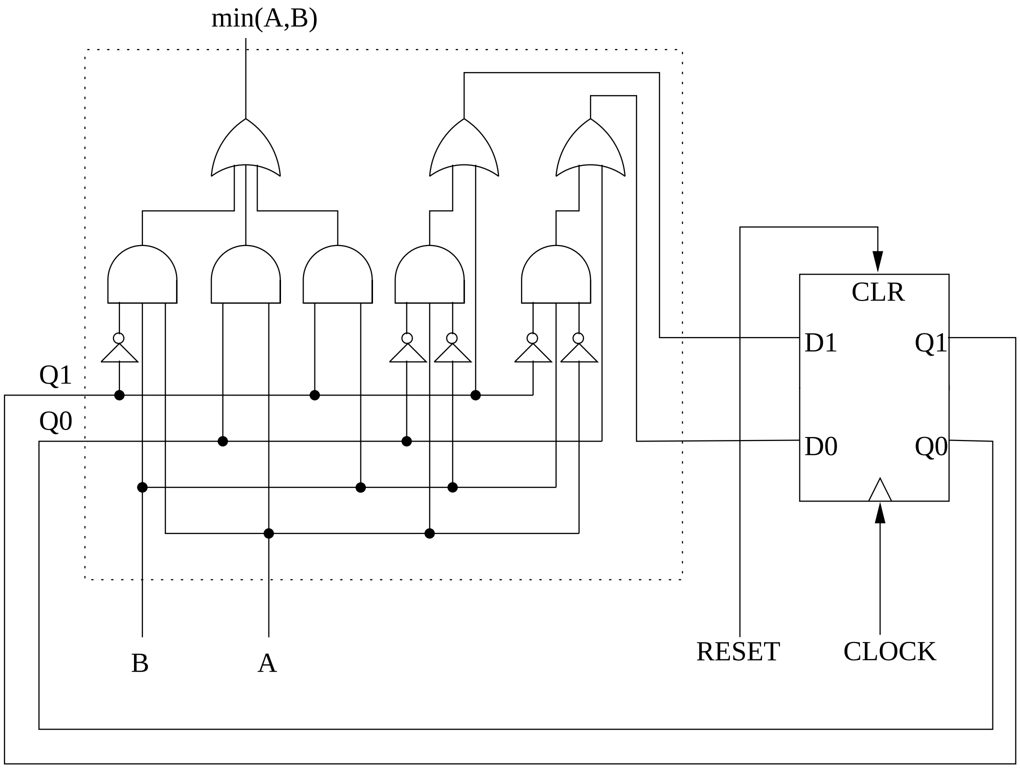


Figure 8: Logic diagram for the Mealey finite state machine to calculate $\min(A, B)$.