Problem 1
a) Initial conditions: assume COUNT is high, D₀ and D₁ are low.

\[
\begin{align*}
Q₀ & \quad T_{\text{reset}} \\
D₁ & \quad T_{\text{read}} (\text{DO}) \\
Q₁ & \quad T_{\text{do}} (\text{DO}) \\
D₀ & \quad T_{\text{write}} (\text{WRITE})
\end{align*}
\]

\[
T_{\text{reset}} + T_{\text{read}} + T_{\text{write}} + T_{\text{do}} = T_{\text{min}} + T_{\text{max}} (T_{\text{reset}} + T_{\text{write}})
\]

So, maximum clock speed is \( \frac{6.25 \text{ns}}{161 \text{ MHz}} \).

b) Initial conditions: assume all signals are low. We are looking for setup time violations for D₀ or D₁.

\[
\begin{align*}
D₀ & \quad T₁ \\
D₁ & \quad T₂
\end{align*}
\]

So, assertion of COUNT between \((T₁/T₂) + T₁ [6 \text{ ns} < T₂] \) causes a setup time violation for D₀ and between \((T₁/T₂) + T₂ [8.8 \text{ ns} < T₂] \) for D₁.

c) chance of violation for a flip flop: \( T_{\text{setup}}/T_{\text{period}} = 0.8 \text{ ns/50 ns} = 1.6\% \)

chance of violation for the whole system: \( 2 \times T_{\text{setup}}/T_{\text{period}} = 3.2\% \)

(The probability of violation for the whole system is double that of a single flip flop because the violations in part b don’t overlap. So, the probabilities simply add up.)

Problem 2. FIFO Design

a) Data path

\[
\begin{align*}
\text{CE} & \quad \text{DATA} \\
\text{GR} & \quad \text{READ/WR} \\
\text{WR} & \quad \text{WREN} \\
\text{RESET} & \quad \text{EN} \\
\text{WRITE} & \quad \text{WRITESEL} \\
\text{READ} & \quad \text{READPTR} \\
\text{LOAD} & \quad \text{LOAD} \\
\text{DOUT} & \quad \text{DOUT}[7:0]
\end{align*}
\]

\[
\begin{align*}
\text{CLK} & \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \\
\text{WRITE} & \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 2 \quad 2 \quad 2 \\
\text{READ} & \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1
\end{align*}
\]

Assumption: WREN, REN are aligned with even clock edges. Use first cycle to read, 2nd cycle to write.
Problem 2

Notes

1) WE = WRITESEL = WRITECNT = WRITE

2) For simplicity & because of time delays in output rep on RAM, do read first if RD-EN is asserted.

3) #read ops × write ops → empty

4) × write ops = × read ops + S12 ⇒ full

Note: This logic is implicitly implemented by defining an extra state.