CSE140 Midterm2 Review

Xinyuan Wang

05/13/2018
Outline

• Flip-Flops
• Finite State Machines
• Timing and Retiming
D Latch

• Two inputs: $CLK$, $D$
  • $CLK$: controls when the output changes
  • $D$ (the data input): controls what the output changes to

• Function
  • When $CLK = 1$, $D$ passes through to $Q$ (the latch is transparent)
  • When $CLK = 0$, $Q$ holds its previous value (the latch is opaque)

• Avoids invalid case when $Q \neq \text{NOT } \bar{Q}$
D Flip-Flop

• Two inputs: CLK, D

• Function
  o The flip-flop “samples” D on the rising edge of CLK
    - When CLK rises from 0 to 1, D passes through to Q
    - Otherwise, Q holds its previous value
  o Q changes only on the rising edge of CLK

• A flip-flop is called an edge-triggered device because it is activated on the clock edge
D-Latch vs. D Flip-Flop

CLK

D Q

Q

D Q

D-latch

D-FF

CLK

D
D-Latch vs. D Flip-Flop
Mealy and Moore Machines

- Mealy Machine: general
  - $y_i(t) = f_i(X(t), S(t))$
- Moore Machine: Output is independent of current input
  - $y_i(t) = f_i(S(t))$
  - $S_i(t + 1) = g_i(X(t), S(t))$
- How to convert Mealy to Moore machine?
Conversion from Mealy to Moore

• Algorithm
  1. Identify distinct (NS, y) pair
  2. Replace each distinct (NS, y) pair with distinct new states
  3. Insert rows of present state = new states
  4. Append each present state with its output y
Conversion from Mealy to Moore

- Example: Q4 from Mid2Wi16

A pattern recognizer produces output \( y = 1 \) when \( x(t - 4, t) = 01010 \)

Mealy Machine

<table>
<thead>
<tr>
<th>PS/Input</th>
<th>( X = 0 )</th>
<th>( X = 1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_0 )</td>
<td>( S_1,0 )</td>
<td>( S_0,0 )</td>
</tr>
<tr>
<td>( S_1 )</td>
<td>( S_1,0 )</td>
<td>( S_2,0 )</td>
</tr>
<tr>
<td>( S_2 )</td>
<td>( S_3,0 )</td>
<td>( S_0,0 )</td>
</tr>
<tr>
<td>( S_3 )</td>
<td>( S_1,0 )</td>
<td>( S_4,0 )</td>
</tr>
<tr>
<td>( S_4 )</td>
<td>( S_3,1 )</td>
<td>( S_0,0 )</td>
</tr>
</tbody>
</table>
Conversion from Mealy to Moore

Mealy Machine

• Algorithm
  1. Identify distinct (NS, y) pair

<table>
<thead>
<tr>
<th>PS/Input</th>
<th>X = 0</th>
<th>X = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>S₀</td>
<td>S₁,₀</td>
<td>S₀,₀</td>
</tr>
<tr>
<td>S₁</td>
<td>S₁,₀</td>
<td>S₂,₀</td>
</tr>
<tr>
<td>S₂</td>
<td>S₃,₀</td>
<td>S₀,₀</td>
</tr>
<tr>
<td>S₃</td>
<td>S₁,₀</td>
<td>S₄,₀</td>
</tr>
<tr>
<td>S₄</td>
<td>S₃,₁</td>
<td>S₀,₀</td>
</tr>
</tbody>
</table>
Conversion from Mealy to Moore

Mealy Machine

- **Algorithm**
  1. Identify distinct (NS, y) pair
  2. Replace each distinct (NS, y) pair with distinct new states
  3. Insert rows of present state = new states

<table>
<thead>
<tr>
<th>PS/Input</th>
<th>$X = 0$</th>
<th>$X = 1$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>$S_1$</td>
<td>$S_0$</td>
<td>0</td>
</tr>
<tr>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_2$</td>
<td>0</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$S_3$</td>
<td>$S_0$</td>
<td>0</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$S_1$</td>
<td>$S_4$</td>
<td>0</td>
</tr>
<tr>
<td>$S_4$</td>
<td>$S_5$</td>
<td>$S_0$</td>
<td></td>
</tr>
<tr>
<td>$S_5$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Conversion from Mealy to Moore

Moore Machine

- Algorithm
  1. Identify distinct (NS, y) pair
  2. Replace each distinct (NS, y) pair with distinct new states
  3. Insert rows of present state = new states
  4. Append each present state with its output y

<table>
<thead>
<tr>
<th>PS/Input</th>
<th>$X = 0$</th>
<th>$X = 1$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>$S_1$</td>
<td>$S_0$</td>
<td>0</td>
</tr>
<tr>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_2$</td>
<td>0</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$S_3$</td>
<td>$S_0$</td>
<td>0</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$S_1$</td>
<td>$S_4$</td>
<td>0</td>
</tr>
<tr>
<td>$S_4$</td>
<td>$S_5$</td>
<td>$S_0$</td>
<td>0</td>
</tr>
<tr>
<td>$S_5$</td>
<td>$S_1$</td>
<td>$S_4$</td>
<td>1</td>
</tr>
</tbody>
</table>
Implementation

✓ Given: State diagram
  → Circuit implementation?
  ○ Example: Lec8 and HW3 “pattern recognizer”

✓ Given: Circuit implementation
  → Input output relation?
Implementation

✓ **Given: State diagram**
  - Derive the state table and assign states
  - Excitation table
  - Equations (K-maps if needed)
  - Circuit implementation
Examples: given a state diagram

• Step 1: Derive state table and assign states

  ○ How many binary variables are needed to encode the states?

<table>
<thead>
<tr>
<th>PS/Input</th>
<th>$X = 0$</th>
<th>$X = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>$S_0,0$</td>
<td>$S_1,0$</td>
</tr>
<tr>
<td>$S_1$</td>
<td>$S_0,0$</td>
<td>$S_2,0$</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$S_0,1$</td>
<td>$S_2,0$</td>
</tr>
</tbody>
</table>
Examples: given a state diagram

• Step 1: Derive state table and assign states

How many binary variables are needed to encode the states?

2 bits

<table>
<thead>
<tr>
<th>State</th>
<th>PS/Input</th>
<th>X = 0</th>
<th>X = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>S₀</td>
<td>S₀, 0</td>
<td>S₁, 0</td>
<td></td>
</tr>
<tr>
<td>S₁</td>
<td>S₀, 0</td>
<td>S₂, 0</td>
<td></td>
</tr>
<tr>
<td>S₂</td>
<td>S₀, 1</td>
<td>S₂, 0</td>
<td></td>
</tr>
</tbody>
</table>
Examples: given a state diagram

**Step 2: Excitation table**

\[ Q_1 Q_0 \]

- \( S_0 \Rightarrow 0\ 0 \)
- \( S_1 \Rightarrow 0\ 1 \)
- \( S_2 \Rightarrow 1\ 0 \)

<table>
<thead>
<tr>
<th>PS/Input</th>
<th>( X = 0 )</th>
<th>( X = 1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_0 )</td>
<td>( S_0,0 )</td>
<td>( S_1,0 )</td>
</tr>
<tr>
<td>( S_1 )</td>
<td>( S_0,0 )</td>
<td>( S_2,0 )</td>
</tr>
<tr>
<td>( S_2 )</td>
<td>( S_0,1 )</td>
<td>( S_2,0 )</td>
</tr>
</tbody>
</table>

- Excitation table

<table>
<thead>
<tr>
<th>( Q_1(t) )</th>
<th>( Q_0(t) )</th>
<th>( X(t) )</th>
<th>( Q_1(t + 1) )</th>
<th>( Q_0(t + 1) )</th>
<th>( y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>( X )</td>
<td>( X )</td>
<td>( X )</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>( X )</td>
<td>( X )</td>
<td>( X )</td>
</tr>
</tbody>
</table>
Examples: given a state diagram

- **Step 3:** Derive equations
  - Use *D Flip-Flop*: $Q(t + 1) = D(t)$
  - Use K-maps

Mealy Machine

$$\begin{array}{cccccccc}
Q_1(t) & Q_0(t) & X(t) & D_1(t) & D_0(t) & Q_1(t + 1) & Q_0(t + 1) & y \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & X & X & X & X & X \\
1 & 1 & 1 & X & X & X & X & X \\
\end{array}$$

**$D_1(t)$**

$$\begin{array}{cccc}
X(t)/Q_1(t)Q_0(t) & 00 & 01 & 11 & 10 \\
0 & 0 & 0 & X & 0 \\
1 & 0 & 1 & X & 1 \\
\end{array}$$

$$D_1(t) = X(t)Q_0(t) + X(t)Q_1(t)$$
Examples: given a state diagram

- **Step 3:** Derive equations
  - Use D Flip-Flop: \( Q(t+1) = D(t) \)
  - K-maps

\[
D_0(t) = X(t)Q'_0(t)Q'_1(t)
\]

\[
y(t) = X'(t)Q_1(t)
\]

<table>
<thead>
<tr>
<th>( X(t)/Q_1(t)Q_0(t) )</th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>X</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( Q_0(t) )</th>
<th>( Q_1(t) )</th>
<th>( X(t) )</th>
<th>( D_0(t) )</th>
<th>( D_1(t) )</th>
<th>( Q_0(t+1) )</th>
<th>( Q_1(t+1) )</th>
<th>( y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

\( D_0(t) \) output

\( y = X'(t)Q_1(t) \)
Examples: given a state diagram

- **Step 4**: Circuit implementation
  - Use D Flip-Flop: $Q(t+1) = D(t)$

  \[
  D_1(t) = X(t)Q_0(t) + X(t)Q_1(t)
  \]

  \[
  D_0(t) = X(t)Q'_0(t)Q'_1(t)
  \]

  \[
  y = X'(t)Q_1(t)
  \]

Don’t forget the label!
Implementation

✓ Given: Circuit implementation
  • Excitation table
  • Identify states and derive a state table
  • State diagram
  • Input output relation
Example: given a circuit implementation

• **Step 1:** Excitation table

![Excitation Table Diagram]

4. (Finite State Machine Specification) Analyze the following circuit.

4.1 Write the transition (excitation) table (8 points).

• **D Flip-Flop:** $Q(t+1) = D(t)$
  - $D(t)$ can be find with current input $X(t)$ and current state $Q0(t)$, $Q1(t)$
Examples: given a circuit implementation

• **Step 1: Excitation table**
  - **State Equations**
    
    \[
    Q_0(t + 1) = XQ_1(t) + XQ_0(t)'
    \]
    
    \[
    Q_1(t + 1) = XQ_1(t) + XQ_0(t)
    \]
    
    \[
    M = X'Q_1(t)Q_0(t)
    \]
  
  - **Excitation table**

<table>
<thead>
<tr>
<th>(Q_1(t))</th>
<th>(Q_0(t))</th>
<th>(X)</th>
<th>(Q_1(t + 1))</th>
<th>(Q_0(t + 1))</th>
<th>(M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Examples: given a circuit implementation

• Step 2: Identify states and derive state table

\[
\begin{align*}
Q_1Q_0 \\
S_0 & \leftarrow 00 \\
S_1 & \leftarrow 01 \\
S_2 & \leftarrow 10 \\
S_3 & \leftarrow 11
\end{align*}
\]

\[
\begin{array}{cccccc}
\text{PS/Input} & X = 0 & X = 1 \\
\hline
S_0 & S_0,0 & S_1,0 \\
S_1 & S_0,0 & S_2,0 \\
S_2 & S_0,0 & S_3,0 \\
S_3 & S_0,1 & S_3,0 \\
\end{array}
\]

\[
\begin{array}{cccccc}
Q_1(t) & Q_0(t) & X & Q_1(t + 1) & Q_0(t + 1) & M \\
\hline
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 \\
\end{array}
\]
Examples: given a circuit implementation

• Step 3: State diagram

<table>
<thead>
<tr>
<th>PS/Input</th>
<th>$X = 0$</th>
<th>$X = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>$S_0,0$</td>
<td>$S_1,0$</td>
</tr>
<tr>
<td>$S_1$</td>
<td>$S_0,0$</td>
<td>$S_2,0$</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$S_0,0$</td>
<td>$S_3,0$</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$S_0,1$</td>
<td>$S_3,0$</td>
</tr>
</tbody>
</table>
Examples: given a circuit implementation

• Step 4: Input output relation

<table>
<thead>
<tr>
<th>cycle</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>state</td>
<td>$S_0$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>cycle</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>state</td>
<td>$S_0$</td>
<td>$S_1$</td>
<td>$S_2$</td>
<td>$S_3$</td>
<td>$S_0$</td>
<td>$S_0$</td>
<td>$S_0$</td>
<td>$S_1$</td>
<td>$S_2$</td>
<td>$S_3$</td>
</tr>
<tr>
<td>$M$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Sequential Networks: setup and hold time

- **Setup time** $t_{\text{setup}}$ Time *before* the clock edge that data must be stable (i.e. not change)
- **Setup time violation** This occurs if the input signal $D$ does not settle *(set up)* to the stable value at least $t_{\text{setup}}$ *before* the clock edge.
- **Hold time** $t_{\text{hold}}$ Time *after* the clock edge that data must be stable
- **Hold time violation** This occurs if the input signal $D$ does not remain unchanged *(hold)* for at least $t_{\text{hold}}$ *after* the clock edge.
Sequential Networks: Timing Constraint (without skew)

• Setup Time Constraint

\[ T_c \geq t_{pcq} + t_{pd} + t_{setup} \]

Longest delay from CLK1 to CLK2
Sequential Networks: Timing Constraint (without skew)

- **Hold Time Constraint**

\[ t_{ccq} + t_{cd} \geq t_{hold} \]

Shortest delay from CLK1 to CLK2
Sequential Networks: Timing Constraint (with skew)

- **Time Skew**
  - Difference between two clock edges
  - \( t_{\text{skew}} = \text{CLK2} - \text{CLK1} \)

- **Setup Time Constraint**
  - \( \text{CLK2} \) is earlier than \( \text{CLK1} \) \( (t_{\text{skew}} < 0) \)
  - \[ T_c \geq t_{\text{pcq}} + t_{\text{pd}} + t_{\text{setup}} - t_{\text{skew}} \]

- **Hold Time Constraint**
  - \( \text{CLK2} \) is later than \( \text{CLK1} \)
  - \[ t_{\text{ccq}} + t_{\text{cd}} > t_{\text{hold}} + t_{\text{skew}} \]
Sequential Networks: Timing Constraint (with skew)

• **Time Skew**
  - Difference between two clock edges
  - $t_{skew} = \text{CLK2} - \text{CLK1}$

• **Setup Time Constraint**
  - CLK2 is earlier than CLK1 ($t_{skew} < 0$)
  
  \[ T_c \geq t_{pcq} + t_{pd} + t_{setup} - t_{skew} \]

• **Hold Time Constraint**
  - CLK2 is later than CLK1
  
  \[ t_{ccq} + t_{cd} > t_{hold} + t_{skew} \]
Example: Q2 from mid2 sp17

- Propagation delay and contamination delay

- Logic gate (AND, OR, XOR)
  - $t_{pd} = 25ps$
  - $t_{cd} = 15ps$

- Flip-Flop
  - $t_{pcq} = 30ps$
  - $t_{ccq} = 20ps$
  - $t_{setup} = 20ps$
  - $t_{hold} = 30ps$

Maximum frequency?
Example: Timing (Without Clock Skew)

• Find the longest path of combinational logic
  \[ t_{pd}(max) = 4 \times t_{pd}(logic\ gate) \]

• Setup Time Constraint
  \[ T_c \geq t_{pcq} + t_{pd} + t_{setup} \]
  \[ T_c \geq (30 + 4 \times 25 + 20) = 150ps \]
  the maximum frequency \( \frac{1}{T} = 6.67GHz \)

• Hold Time Constraint
  \[ t_{cd}(min) = t_{cd}(logic\ gate) = 15ps \]
  \[ t_{ccq} + t_{cd}(min) = 20 + 15 = 35ps \geq t_{hold} \]
  \[ = 30ps \]
Example: Timing (With Clock Skew)

• Hold Time Constraint
  \[ t_{ccq} + t_{cd(min)} = 35\text{ps} \geq t_{\text{hold}} + t_{\text{skew}} = (30 + \Delta)\text{ps} \]
  \[ t_{\text{skew}} = 5\text{ps} \quad \text{CLK2 Later than CLK1 !!!} \]

• Setup Time Constraint
  \[ T_c + t_{\text{skew}} \geq t_{pcq} + t_{pa} + t_{\text{setup}} \]
  \[ T_c \geq (150 - 5) = 145\text{ps} \]
  the maximum frequency \[ \frac{1}{T} = 6.90\text{GHz} \]
Good luck!