MS (CE Major) Exam Solution

1. Static Timing Analysis (STA) (50pt)

A.

B. \( T_{\text{clock}} \geq T_{\text{CK} - Q} + T_{\text{dina}} + T_{\text{setup}} \)
Minimum clock cycle time = 0.5 + 15 + 0.5 = 16 (ns)

C. \( T_{\text{clock}} \geq T_{\text{CK} - Q} + T_{\text{dina}} + T_{\text{setup}} + T_{\text{skew}} \)
\( T_{\text{skew}} = T_{\text{launch}} - T_{\text{capture}} \)

\( r_w = 1 \), \( c_w = 2 \text{fF} \), \( C_{\text{CK}} = 1 \text{pF} \),
\( T_{\text{CK} - Q} = 0.5 \text{ns} \), \( T_{\text{setup}} = 0.5 \text{ns} \)

- \( T_{\text{capture}} = T_{A - C} = R_w (C_w + C_{\text{CK}}) = 2 \text{ (ns)} \)

\( R_w = c_w \cdot L_2 \)
\( = 1 \times 1000 \) / 2 \( \text{fF} \)

\( C_w = c_w \cdot L_2 / 2 \)
\( = 2 \times 1000 / 2 \text{fF} \)

\( C_{\text{CK}} = 1 \text{pF} \)
\[-T_{\text{srchck}} = T_{d-B} = R_w (C_w + C_{CK}) = 6 \text{ (ns)}\]

Minimum clock cycle time = 0.5 + 15 + 0.5 + (6 - 2) = 20 (ns)

2. Simple power network design (20pt)

\[R_{10um} = R_{\text{shunt}} \frac{L}{W} = 0.1 \frac{10}{W} = \frac{1}{W}\]

\[\text{Max(\text{IR-Drop})} = R_{10um} \sum_{i=1}^{100} i = \frac{100 \cdot 101}{2W} = 120 (mV)\]

\[W \approx 42 \mu m\]

3. Explain the following (30pt)

A. Gate delay depends on drive current, input slew and output load capacitance. Timing slack depends on the arrival time and required time. Typical techniques are as follows. (However, any techniques that can effectively reduce input slew or output load capacitance, or improve driving strength, or reduce arrival time or increase required time, can be answers.)

i. **Resizing**: in general, larger cells having larger width transistors can charge or discharge load capacitance rapidly. However, as shown in the following figure, larger cells can have higher delay when the load capacitance is small. In addition, larger cells consume larger power and area. (In the following figure, cell “C” has smaller delay for the given load capacitance 0.7.)
ii. **Cloning**: cloning is performed as in the following figure. Driver cell is cloned, and each driver cell has lower fanout load. Hence, the driver cells’ delay can be reduced. However, this technique increases area and power, and if cells driving node “a” or “b” are small, those cells’ delay can increase, since load capacitance increases by twice.

iii. **Buffering**: when a cell drives many fanout cells, the cell’s delay can be excessive. When a buffer is inserted as shown in the following figure, the Nand gate delay can decrease, since fanout load for “f”, “g”, and “h” are unseen from the Nand gate. However, this technique can increase area and power of circuits, and the delay from “a” or “b” to “f”, “g”, or “h” can increase.
iv. **Fanin tree redesign**: fanin tree can be redesigned considering arrival time of inputs "a", "b", "c" and "d". The main idea is to reduce delay of the critical timing path, i.e., a path "a" to "e".

```
\begin{array}{c}
\text{Arr}(a)=4 \\
\text{Arr}(b)=3 \\
\text{Arr}(c)=1 \\
\text{Arr}(d)=0 \\
\end{array}
\begin{array}{c}
\text{Arr}(e)=5 \\
\end{array}
```

v. **Fanout reclustering**: fanout tree can be redesigned according to the timing criticality. The main idea is to reduce the number of logic stages of the most timing critical paths by reclustering.
vi. **Logic decomposition (restructuring, remapping):** complex cells in general have lower drive strength based on the logical effort theory. Hence, logic decomposition of a complex cell to a group of simple cells can have smaller delay. This method can increase area and power, and some of delay of the decomposed circuits can be larger than the original complex cell.

vii. **Input reordering (swapping):** this technique swaps inputs without changing functionality. Inputs that have larger arrival time are connected to nodes that have larger required time.
Simple sorting on arrival times and delay works

viii. **Repeater insertion**: when a cell drives long interconnect, buffers can be inserted on the long interconnect.

ix. Other optimization techniques: **retriming, useful skew, multi-vth, etc.**

B. Correct sequence of design steps
   
i. **Design specification**: defines inputs, outputs and the desired functionality of designs

ii. **RTL design**: based on the design specification, design's behavior is described in terms of the flow of signals between registers, using hardware description languages (HDLs), such as Verilog or VHDL

iii. **Logic synthesis**: the functions described in HDLs are mapped to Boolean logic gates. Timing or area is optimized during logic synthesis

iv. **Floorplan**: very first stage of physical design. Floorplan defines size and shape of the design to be physically implemented, including locations of logic blocks and input/output terminals

v. **Power plan**: constructs a power delivery network from the power sources to logic blocks

vi. **Placement**: determines locations of logic gates considering floorplan constraints and timing constraints. Logic optimization can also be performed

vii. **Clock tree synthesis**: constructs a clock network that connect the clock sources to registers

viii. **Routing**: logical connectivity between logic gates are implemented (=routed) using (metal) interconnects

ix. **Tapeout**: information of the transistors and interconnects in placed and routed designs are transferred to manufacturers