Very Large Scale Integration (VLSI)

Lecture 6

Dr. Ahmed H. Madian
Ah_madian@hotmail.com

Contents

- FPGA Technology
  - Programmable logic Cell (PLC)
    - Mux-based cells
    - Look up table
    - PLA
  - Programmable interconnection
    - Antifuse
    - SRAM
    - EPROM
  - Placement and Routing
    - Min-Cut
    - Simulated Annealing
Field Programmable Gate Array (FPGA)

- Programming technology
  - Programmable logic cells
  - Programmable interconnect
    - Anti fuse (One time programmable, ex. Actel)
    - Static RAM (volatile, ex. Xilinx)
    - EPROM or EEPROM (non-volatile, ex. Altera)
  - Programmable I/O’s

Programmable Logic Cells

- The logic cells could be programmed to realize a logic function the main blocks are:
  - Multiplexer based (ACTEL)
  - Look-up tables (XILINX)
  - AND-OR array (ALTERA)
Programmable Logic Cells

- Multiplexer Based logic cells
  - The logic function could be written using shanon’s expansion theorem:

\[
F(A, B, C, \ldots) = A \cdot F_{A=1} + \bar{A} \cdot F_{A=0}
\]

![Diagram](image)

- Realize the following logic function using Multiplexer Based logic cells

\[
F = A \cdot B + C \cdot \bar{B} + D
\]
Programmable Logic Cells

- Multiplexer Based logic cells
  \[ F = A \cdot B + C \cdot \overline{B} + D \cdot (B + \overline{B}) \]
  where \( B + \overline{B} = 1 \)
  \[ F = B \cdot (A + D) + \overline{B} \cdot (C + D) \]
- \( F_{B=1} = A+D = A(1) + \overline{A}(D) \)
- \( F_{B=0} = C+D = C(1) + \overline{C}(D) \)

Programmable Logic Cells

- Multiplexer Based logic cells
  - Combinational module (C-module)
  - Sequential module (S-module)
Programmable Logic Cells

- Layout of the 4x1 Mux using TG technology.
- Not always compact layout, but you can make a nice Boolean Unit like in table below.

<table>
<thead>
<tr>
<th>I₃</th>
<th>I₂</th>
<th>I₁</th>
<th>I₀</th>
<th>Out(o)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Zero</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>AND</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>OR</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>XOR</td>
</tr>
</tbody>
</table>

Look-Up tables (LUT)

- LUT mainly depends on RAM blocks
- We could realize any logic function from truth table using RAM
  - This is done by mapping the inputs to the address bus
  - The output is mapped to the data bus
FPGA (SLICE)

Slices (grouped into CLBs = Configurable Logic Blocs)
- Contain combinatorial logic and register resources
  - For example each Spartan™-3 CLB contains four slices:
    - Slices are grouped in pairs
      - Left-hand SLICEM (Memory)
        - LUTs can be configured as memory or SRL16
      - Right-hand SLICEL (Logic)
        - LUT Switch Matrix Slice can be used as logic only.
Programmable Logic Arrays (And-OR-Plane) (PLA)

- It usually constructed directly from minimized Sum-Of-Product logic equations:
- the rows represent the minterms of the equations, the “input” columns form the minterms and the “output” columns form the sums.
- Note that with multiple output columns, minterm sharing between the outputs happens naturally

Address decoder implemented as AND (= NOR)

PLA Folding

- PLAs can be sparse, i.e., only a few of the possible connections in either plane may be made. (AND plane can only have 50%)
- If we allow input and outputs to come from both above and below then we may be able to fold two columns into one if the rows they use don’t overlap. This may require rearranging the rows to minimize overlap and hence maximize folding possibilities.
Multiple Input Encoding

- On the previous slide, it was noted that the AND plane can have at most 50% of its connections programmed.
- To improve the utilization of the input columns, consider encoding the 4 columns used to transmit the two input literals and their complements with some more useful functions of the two literals. For example:

```
A  A  B  B
define AB  AB  AB  AB
  A  A  B  B
  A  A  B  B

PLA
```

- How to expand the PLA size?
- Problems:
  - \( n \) times the number of inputs and outputs requires \( n^2 \) as much chip area -- too costly
  - logic gets slower as number of inputs to AND array increases
- Solution use multiple PALs with small number of inputs
Complex Programmable Logic Device (CPLD)

- How do we get more gates? We could put several PALs on one chip and put an interconnection matrix between them.

Contents

- FPGA Technology
  - Programmable logic Cell (PLC)
    - Mux-based cells
    - Look up table
    - PLA
  - Programmable interconnection
    - Antifuse
    - SRAM
    - EPROM
  - Placement and Routing
    - Min-Cut
    - Simulated Annealing
Anti fuse

- One time programmable
- The channel routing uses dedicated rectangular areas of fixed size within the chip called wiring channels
- Within the horizontal or vertical channels wires run horizontally or vertically, respectively, within tracks
- Actel divides the fixed interconnect wires within each channel into various lengths or wire segments
- The designer then programs the interconnections by blowing antifuses and making connections between wire segments

RC Delay in Antifuse Connections

Elmore Delay

- (a) A four-antifuse connection. L0 is an output, L1 and L3 are horizontal tracks, L2 is a long vertical track (LVT), and L4 is an input
- (b) An RC-tree model. Each antifuse is modeled by a resistance and each interconnect segment is modeled by a capacitance.
- Interconnect delay grows quadratically ($x^2$) as we increase the interconnect length and the number of antifuses, n
SRAM Cell

- Static Ram reprogrammed according to your bit file, it holds the state until new design is downloaded
  - ‘1’ means the switch is on
  - ‘0’ means switch is off
- SRAM is reprogrammable, volatile.

Interconnect delay in FPGA

- The vertical lines and horizontal lines run between CLBs.
- The general-purpose interconnect joins switch boxes (also known as magic boxes or switching matrices).
- The long lines run across the entire chip. It is possible to form internal buses using long lines and the three-state buffers that are next to each CLB.
- The direct connections (not used on the XC4000) bypass the switch matrices and directly connect adjacent CLBs.
- The Programmable Interconnection Points (PIP’s) are programmable pass transistors that connect the CLB inputs and outputs to the routing network.
- The bi-directional (BIDI) interconnect buffers restore the logic level and logic strength on long interconnect paths.
EPROM, EEPROM and Flash

- Reprogrammable nonvolatile memory.
- These memories use a second layer of polysilicon to form a floating gate between the primary gate and the channel.
- The floating gate is a good conductor, but it is not attached to anything.
- Applying a high voltage for the upper gate causes electrons to jump through the thin oxide onto floating gate through the processes called avalanche injection.
- Injecting electrons includes a negative voltage on the floating gate, which increasing threshold voltage of the transistor to the point that it is always OFF.

![Diagram of EPROM structure](image)

Contents

- FPGA Technology
  - Programmable logic Cell (PLC)
    - Mux-based cells
    - Look up table
    - PLA
  - Programmable interconnection
    - Antifuse
    - SRAM
    - EPROM
  - Placement and Routing
    - Min-Cut
    - Simulated Annealing
### Placement

- The process of arranging the circuit components on a layout surface.
- Inputs: A set of fixed modules, a netlist.
- Goal: Find the best position for each module on the chip according to appropriate cost functions.
  - **Considerations:** routability/channel density, wirelength, cut size, performance, thermal issues, I/O pads.

### Estimation of Wirelength

- Semi-perimeter method: Half the perimeter of the bounding rectangle that encloses all the pins of the net to be connected. Most widely used approximation!
- Complete graph: $\text{wire length} = \frac{2}{n} \sum_{(i,j)} \text{distance}(i,j)$ where I and j are nodes to be connected and n number of edges.
- Minimum chain: Start from one vertex and connect to the closest one, and then to the next closest, etc.
- Source-to-sink connection: Connect one pin to all other pins of the net. Not accurate for uncongested chips.
- Steiner-tree approximation: Computationally expensive.
- Minimum spanning tree
Example for wirelength estimation methods

- Partitioning-Based Placement (min-cut)
- Simulated Annealing Placement
- Quadratic Placement
- Hybrid and Hierarchical Placement
Min-Cut Placement

- Breuer: A class of min-cut placement algorithms," DAC-77.
  - Quadrature: suitable for circuits with high density in the center.
  - Bisection: good for standard-cell placement.
  - Slice/Bisection: good for cells with high interconnection on the periphery.

Algorithm for MIN-CUT Placement

Algorithm: Min Cut Placement(N; n;C)
/* N: the layout surface */
/* n: # of cells to be placed */
/* n0: # of cells in a slot */
/* C: the connectivity matrix */
begin
if (n≤n0) then PlaceCells(N; n;C);
else
  (N1;N2) ← CutSurface(N);
  (n1;C1), (n2;C2) ← Partition(n;C);
  Check for Min Cut Placement(N1; n1;C1);
  Check for Min Cut Placement(N2; n2;C2);
end
Placement Example

Partition the shown circuit using MIN-CUT Quadrature method

solution
Min-Cut Placement with Terminal Propagation

- Drawback of the original min-cut placement: Does not consider the positions of terminal pins that enter a region.
- What happens if we swap (1, 3, 6, 9) and (2, 4, 5, 7) in the previous example?

Min-Cut Placement with Terminal Propagation

- Partitioning must be done breadth-first, not depth-first.

Dr. Ahmed H. Madian-VLSI
MIN- CUT placement

- Pros:
  - Open Cost Function (partitioning cost)
    - Minimize net cut, edge cut etc.
  - It is Move Based, Suitable to Timing Driven Placement

- Cons:
  - Lots of “indifferent“ moves
  - May not work well with some cost functions
  - Multi partitioning

Simulated Annealing

- Stage 1
  - Modules are moved between different rows as well as within the same row.
  - Modules overlaps are allowed.
  - When the temperature is reached below a certain value, stage 2 begins.
- Stage 2
  - Remove overlaps.
  - Annealing process continues, but only interchanges adjacent modules within the same row.
Example

- Solution Space: All possible arrangements of the modules into rows, possibly with overlaps.
- Neighborhood Structure: 3 types of moves
  - M1: Displace a module to a new location.
  - M2: Interchange two modules.
  - M3: Change the orientation of a module.

Assignment 3

- Assignment 3 will be available this week on the website