ASICs
Application Specific Integrated Circuit
ASICS - What are they?

- An ASIC is:
  - An Application Specific Integrated Circuit
  - An ASIC is an IC that is designed to perform a particular, specialized function
  - It is not software programmable, unless it’s a microcontroller
  - It is not a memory chip, but may contain memory
- Examples would include:
  - MPEG decoder
  - Audio processor for Dolby noise reduction
  - Image processor for MRI machine
- How are ASICS used?
  - Many popular electronic devices
  - High volume, cost sensitive
  - High reliability, high performance (mA/Mhz)
ASICS – Types of ASICS

- Full Custom ASIC
  - Transistors are sized, placed and connected one by one
  - Analog parts are done this way
  - Highest performance, longest design time
  - Hand optimized parts of high-end microprocessor
  - Full set of masks needed
ASICs – Types of ASICs

- Standard Cell ASIC
  - Predesigned (standard) cells in a library are used
  - These cells are connected together to form the design
  - Reasonable performance, much shorter design time than full custom
  - Most digital ASICs are designed this way
  - Usually uses HDLs with logic synthesis (verilog, VHDL)
  - Full set of masks needed
ASICs – Types of ASICs

• Gate Array ASIC
  • Predrawn transistors are on a die
  • These transistors can be connected together in any fashion to form cells
  • The cells are connected to form a design
  • Usually digital in nature
  • Lower performance, but very quick to fabricate
  • Only metallization masks are needed
  • Usually uses HDLs with logic synthesis
ASICs – Types of ASICs

- FPGA
  - Fully predesigned silicon, logic functions and interconnects
  - Programmed with a “bit file” to configure logic and interconnects
  - Medium performance, higher power consumption and price
  - Very quick to implement, programmable in minutes
  - Better for more limited volume applications
  - Almost always uses HDLs (verilog, VHDL) plus logic synthesis
  - No DFT structures required
  - Popular for ASIC prototyping

Altera Stratix V GT $12,995

Lattice iCE40 $1.65
ASICs – Considerations

- Time to market
  - Time to market is a primary driver
  - Dramatic increase in profit with shorter times
  - Cutting one month off the schedule can increase profits by as much as 10%
ASICS – Considerations

- Cost
  - Standard cell synthesis can cost $125,000 per license
  - Typical standard cell NRE $50,000 to $200,000
  - If you can’t spend at least 1 million, you are not in the game
  - One bug in a standard cell design can be a disaster
  - A standard cell “re-spin” can take 8 weeks or more can cost a lot of money
  - Really cheap tools are available for FPGA design, with some even free!
  - FPGAs cost anywhere from $2-25,000 each in single quantity
  - FPGA bugs can be fixed at zero cost and in minutes

<table>
<thead>
<tr>
<th>Process (nm)</th>
<th>VDD (v)</th>
<th>Metal Layers</th>
<th>Mask Set Cost ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>65</td>
<td>1.0</td>
<td>9</td>
<td>3,000,000</td>
</tr>
<tr>
<td>90</td>
<td>1.0</td>
<td>9</td>
<td>1,500,000</td>
</tr>
<tr>
<td>130</td>
<td>1.2</td>
<td>7</td>
<td>750,000</td>
</tr>
<tr>
<td>350</td>
<td>3.3</td>
<td>3</td>
<td>40,000</td>
</tr>
</tbody>
</table>

(from 2006)
ASICS – How do you choose?

• 1 – Technical feasibility
  • Can it run fast enough?
  • Is it low enough power?
  • Can it operate at the intended supply voltage?
  • Is the required IP available?

• 2 – Financial Analysis
  • What is the cost to get first prototypes (NRE)
  • What is the price per chip at reasonably high volume?
  • How many will we ship?

• FPGAs generally better choice for low volume production (<10,000 units)
• ASICS will pan out if the chip is planned to sell at high volume (>100,000 units)
ASICs – Example
UBR Block Diagram

Note Title

ASICS – Example

November 5, 2013
ASICs – Example

Counter State Machine

Mealy or Moore?
ASICs – Example

Counter State Machine

**Moore**: output values defined by current state only
module counter_sm (
    input     clk,
    input     reset_n,
    output reg restart,
    output reg restart_mem,
    input     in
);

// state encoding outputs | present state
// restart, restart_mem | ps[3:0]
enum reg [4:0]{
    INIT   = 5'b00_000,
    IN_FOUND = 5'b11_001,
    RESET_DONE = 5'b01_010,
    WAIT_2   = 5'b01_011,
    WAIT_3   = 5'b01_100,
    ALL_DONE = 5'b00_110,
    GO_TO_INIT = 5'b00_111,
    X       = 5'bxx_xxx
} counter_ps, counter_ns;

reg sync;

// advance state machine every clock cycle
always_ff @(posedge clk, negedge reset_n)
begin
    if(~reset_n)
        counter_ps <= INIT;
    else
        counter_ps <= counter_ns;
end

// sync up the sensor input with an extra flip flop
always_ff @(posedge clk, negedge reset_n)
begin
    if(~reset_n)
        sync <= 1'b0;
    else
        sync <= in ;
end

// counter state machine logic
always_comb
begin
    counter_ns = X;
    case (counter_ps)
    INIT:  if (sync) counter_ns = IN_FOUND;
             else counter_ns = INIT;
    IN_FOUND: counter_ns = RESET_DONE;
    RESET_DONE: counter_ns = WAIT_2;
    WAIT_2:   counter_ns = WAIT_3;
    WAIT_3:   counter_ns = ALL_DONE;
    ALL_DONE: if (~sync) counter_ns = GO_TO_INIT;
               else counter_ns = ALL_DONE;
    GO_TO_INIT: counter_ns = INIT;
    endcase
end

counter, restart_mem = counter_ps[4:3];
end
endmodule
Verilog Code – Example

- Create Working Directory
  - Put all your .v or .sv files in a single directory then enter the following in the terminal:
    - vlib work
- Compiling Verilog
  - Once your code is ready to compile use the command:
    - vlog yourcode.sv
- Simulating with Modelsim
  - To launch Modelsim to simulate (you can also use it to compile) type:
    - vsim
- Synthesis with Design Vision
  - To launch Design Vision for synthesis (to see the gate level schematic) type in terminal:
    - design_vision-xg
Why do ECE’s get Halloween and Christmas mixed up?
Enjoy

Why do ECE's get Halloween and Christmas mixed up?

Because 31oct = 25dec