FPGA Familiarization

(Introduction to Field Programmable Gate Arrays)

The International System Safety Society
Tennessee Valley Chapter
May 18, 2016

Charles Fulks
fpga@irtc-hq.com

Approved for public release; distribution is unlimited.

Presentation Outline

1. What is Digital Design?
2. What is an FPGA?
3. Manufacturers and Tools
4. Quality Indicators
   - Development Process
   - Management
   - Technical
5. Concluding Thoughts

Class Information

Abstract

Field Programmable Gate Arrays are becoming ubiquitous is electronics. Many people misunderstand the nature of these devices and confuse their development with software development. This session introduces Field Programmable Gate Array (FPGA) technology and development. This is intended for engineers and management who need to understand FPGAs, but who do not intend to personally develop FPGA designs.
The attendee will leave with a solid foundation of FPGA technology, development process, and management. They will also have basic knowledge of common errors and indicators of design quality (red flags).

Disclaimer

This presentation is not meant as a stand-alone document, and cannot be used effectively without the accompanying verbal discussion.

Introductions

Curriculum Vitae

Charles Fulks leads the FPGA development group for Intuitive Research and Technology Corporation (www.irtc-hq.com). With over 20 years in the embedded / high reliability industry, Charles works with a number of different technologies. However, his focus over the past decade is primarily Field Programmable Gate Arrays (FPGA) and embedded digital design. He has patented FPGA related technology. He holds a MSEE degree from the University of Central Florida and is a Senior Member of the IEEE. Charles has trained numerous design engineers, is a regular speaker at several conferences, and has presented on the topic of FPGA design internationally. He was interviewed for EE Web's Featured Engineer column.

We develop opinions based on our personal experience (and reading). Experience is usually field related. The opinions developed in academia are therefore necessarily different from those developed in a safety critical environment. The opinions expressed in this paper are those of an engineer, experienced in FPGA design in a high reliability field.[a]

[a] Your mileage may vary.

What is Digital Design?

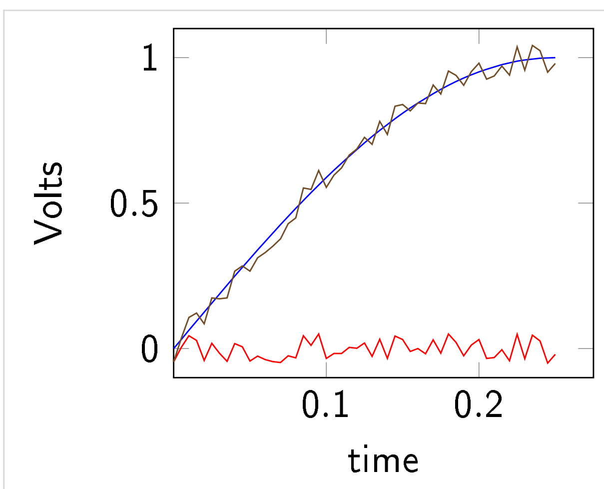
Fundamentals

- What is an FPGA?
  - A place to implement a digital design.
- Then, what is digital design?

This discussion is intended as a brief introduction to the field of digital circuit design; not a comprehensive overview.

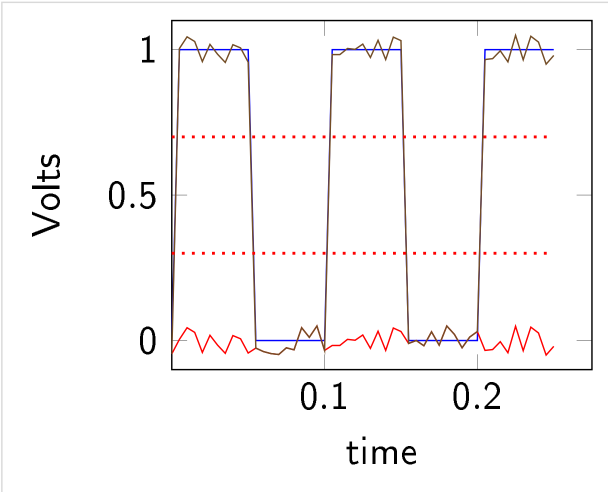
Analog Signals

- "Real world" signals
- Continuous range of values
- Sensitive to noise



Digital Signals

- Digital computer signals
- Discrete (discontinuous) values
- Much less sensitive to noise

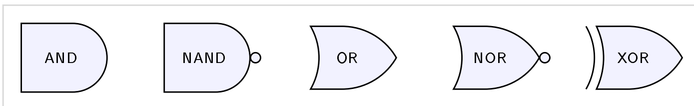


Binary Numbers

2^3	2^2	2^1	2^0	Decimal	Hex
8	4	2	1		
0	0	0	0	0	0
1	0	0	1	9	9
1	0	1	0	10	A
1	1	1	1	15	F

Logic Functions

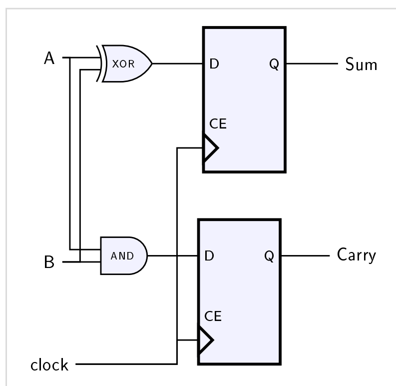
A	B	AND	NAND	OR	NOR	XOR
0	0	0	1	0	1	0
0	1	0	1	1	0	1
1	0	0	1	1	0	1
1	1	1	0	1	0	0



Binary Addition

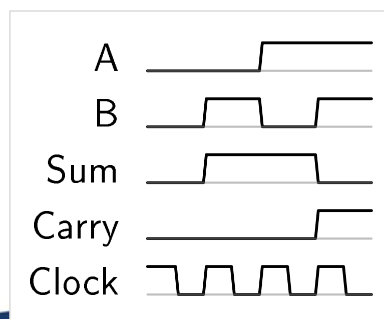
Half-Adder

A	B	A+B	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



Clock Synchronous Circuit

Add registers to control timing



What is Digital Design?

Key Points

- Digital design is the design of circuits that process digital (binary) signals.
- The clock is the “heartbeat” of the digital circuit.

Why digital signals?

It is easier to get an electronic device to switch into one of a number of known states than to accurately reproduce a continuous range of values.

Language of FPGA Technology

Term	Definition
PLD	Programmable Logic Device
CPLD	Complex Programmable Logic Device
FPGA	Field Programmable Gate Array; the most common PLD
SoC	System-on-Chip; a silicon device that includes 1 or more processor cores with the FPGA
Complex Electronics	A term used to describe any electronic device that cannot be comprehensively tested.

Industry Standards

Federal Aviation Administration

RTCA DO-178

"Software Considerations in Airborne Systems and Equipment Certification"

FAA Advisory Circular 20-152 recommends RTCA/DO-254

Excerpt from RTCA/DO-254 Design Assurance Guidance For Airborne Electronic Hardware:

- A hardware item is identified as simple only if a comprehensive combination of deterministic tests and analyses appropriate to the design assurance level can ensure correct functional performance under all foreseeable operating conditions with no anomalous behavior.
- When an item cannot be classified as simple, it should be classified as complex.

Industry Standards

NASA

Assurance Process for Complex Electronics

www.hq.nasa.gov/office/codeq/software/ComplexElectronics/index.htm

- Complex electronics are programmable devices that can be used to implement specific hardware circuits. The devices that are included under the label of complex electronics are: CPLD, FPGA, ASIC...
- In the term complex electronics, the complex adjective is used to distinguish between simple devices, such as off-the-shelf ICs and logic gates, and user-creatable devices.

> "Note that firmware (which is essentially software stored on a read-only device) is not considered complex electronics. The integrated circuit (e.g. EPROM) is simple electronics. The program stored in that device is software, which has a defined assurance process in place."

FPGA Evolution - Overview

We started with hardware

- A new algorithm required re-wiring the system

Then we invented software

- The algorithm is independent from the physical hardware
- Complexity of software is very high with respect to hardware
- Developed a process to ensure quality

Then we invented programmable logic (FPGAs)

- The **digital design** is independent from the physical hardware
- Digital design complexity is on par with software
- Need to follow a process to ensure quality

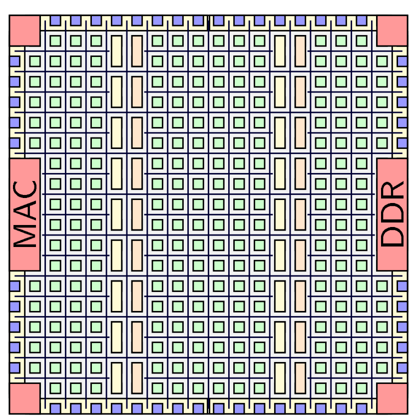
FPGA Device Description

FPGA Characteristics

- An FPGA is a generic, blank digital device
- It has many of each type of element but they are not movable

FPGA Elements

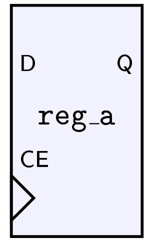
- Logic elements (LE)
- "Banks" of IO Blocks
- Multipliers
- Block RAM
- Clocking resources
- Silicon IP
- Routing resources



FPGA Design Capture

- Schematic *(obsolete)*
- Hardware Description Languages (HDL)

```vhdl
p_reg_a : process( clock )
begin
  if rising_edge( clock ) then
    if ( reset = '1' ) then
      reg_a <= (others=>'0');
    else
      reg_a <= i_reg_a;
    end if;
  end if;
end process p_reg_a;
```



Key Point: FPGA design is not software development; FPGA design is digital design. It requires the digital design skill set.

Case Study — Input Debouncer

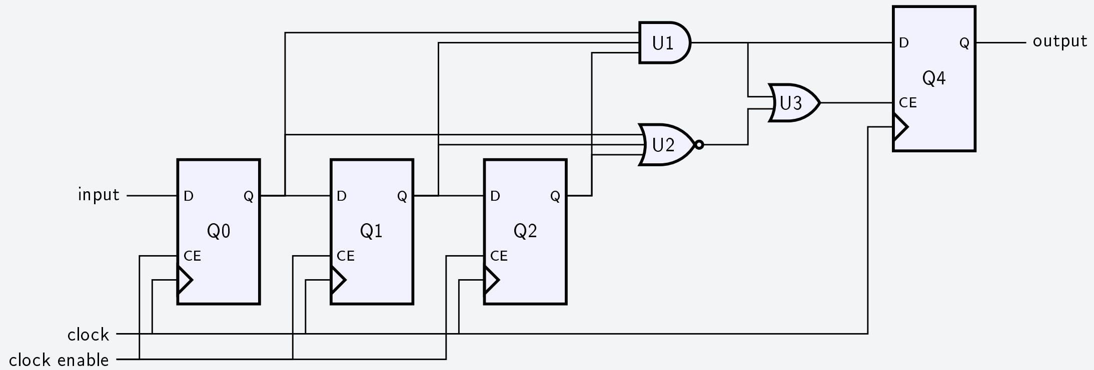
Hardware Description Language (VHDL)

```vhdl
library ieee;
  use ieee.std_logic_1164.all;

entity debounce is
port(
   discrete_in  : in  std_logic;
   discrete_out : out std_logic;
   ce    : in std_logic;
   clock : in std_logic;
   reset : in std_logic
);
end debounce;

architecture debounce_arch of
     debounce is
  signal sample   : std_logic_vector
     (3 downto 1);
  signal all_high : std_logic;
  signal all_low  : std_logic;

begin
  p_sample : process (clock)
  begin
    if rising_edge(clock) then
      if (reset = '1') then
        sample <= (others => '0');
      elsif (ce = '1') then
        sample(1) <= discrete_in;
        sample(2) <= sample(1);
        sample(3) <= sample(2);
      end if;
    end if;
  end process;

  all_high <= '1' when sample = "111
     " else '0'; -- AND Gate
  all_low  <= '1' when sample = "000
     " else '0'; -- NOR Gate

  p_discrete_out : process (clock)
  begin
    if rising_edge(clock) then
      if (reset = '1') then
        discrete_out <= '0';
      elsif (all_high = '1') or (
          all_low = '1') then
        discrete_out <= all_high;
      end if;
    end if;
  end process;
end debounce_arch;
```

Case Study — Input Debouncer

Debouncer Logic Diagram

Debouncer Logic Diagram



Key points

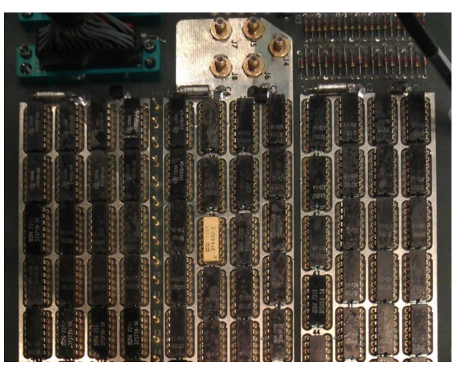
The VHDL text describes a circuit that consists of 4 flip-flops and 3 logic gates.

The synthesizable subset of VHDL describes digital logic circuits.

Case Study — Input Debouncer

How do you realize the debouncer design?

Discrete Logic Circuit Board



Application Specific Integrated Circuit (ASIC)

On a silicon wafer, place transistors that combine to form 4 flip-flops and 3 gates (design and fabricate a custom IC).

- Unchangable → Significant NRE invested in a single product

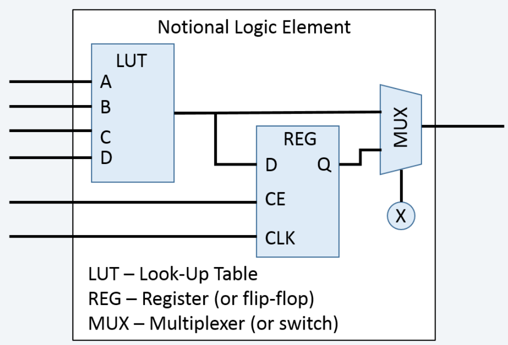
Case Study — Input Debouncer

How do you realize the debouncer design?

Create a generic "Logic Element" (LE)

- Comprised of a 16-bit memory, a flip-flop, and a multiplexer
- The values stored in the memory and control bits determine the function

Q0	Q1	Q2	D	CE
0	0	0	0	1
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



Case Study — Input Debouncer

How do you realize the debouncer design?

Connect logic elements to perform the desired function

- The synthesis tool automates this
  - The designer must be aware of how the HDL is physically implemented
- There are many inputs to the synthesis tool other than HDL
  - Pin assignments
  - Timing information
  - Tool settings
  - etc.

Case Study — Input Debouncer

Synthesis Process

Term	Definition
Synthesize	Generate a netlist from the HDL
Translate	Reduce the design to "primitives" specific to the FPGA manufacturer
Map	Select available logic blocks to implement logic in the netlist
Place and Route (PAR)	Select specific logic blocks on the physical FPGA silicon and select routing resources to connect them

Case Study — Input Debouncer

Key Points

- FPGA design is **digital logic circuit design**
- The FPGA designer must follow a **good design process** to get good results
  - The design must be documented and reviewed prior to HDL coding
- **Functional simulation** is required for every design

A Comparison. . .

Apples and Oranges — Both are grown in orchards, but. . .

- FPGAs and Software are fundamentally different things
- They require fundamentally different skill sets
- The level of design complexity is similar
- They require an almost identical development process
  - Revision control
  - Design phases
  - Review phases
  - Simulation
  - Integration

Manufacturers

Large Manufacturers

- Altera (www.altera.com) — Recently purchased by Intel
- Xilinx (www.xilinx.com)

Smaller Manufacturers

- Atmel (www.atmel.com)
- Lattice (www.latticesemi.com)
- Microsemi (www.microsemi.com) — the artist formerly known as Actel
- QuickLogic (www.quicklogic.com)

Tools

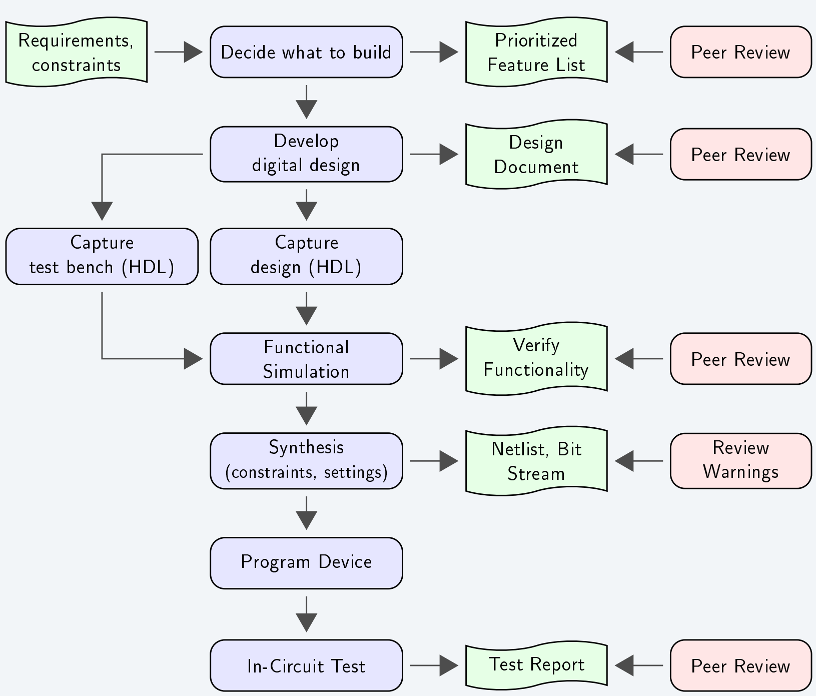
Simulation

- Aldec Active-HDL, Rivera-PRO (www.aldec.com)
- Mentor Graphics ModelSim, Questa
- Xilinx Vivado (www.xilinx.com)

Synthesis

- Altera Quartus (www.altera.com)
- Lattice Diamond (www.latticesemi.com)
- Microsemi Libero (www.microsemi.com)
- Xilinx Vivado (www.xilinx.com)

Notional Minimum Development Process



Quality Indicators

Development Process

How much process?

- Use a level of process appropriate for the criticality of the design — too much process fosters avoidance
- Give the design team the authority to modify the process where needed with appropriate justification and oversight

Key points

- You cannot completely test an FPGA design. Adherence to a good process helps ensure a level of quality.
- There is more to the story than the source code. A process provides a disciplined and documented way to capture critical design artifacts.

Process

Requirements

Key Point

FPGA requirements must be well matched to the FPGA capabilities

"How will I know when I'm done?"

- Technical lead reviewed "prioritized feature list"
- Models for all signal processing algorithms

Requirements change?

- All projects encounter requirements changes
- These adversely affect schedule, cost, and quality
- Closing the loop eliminates requirements changes due to misunderstandings

Process

Documentation

Design document versus “As-Built” document

- “Think, then act” as opposed to “Act, then describe what we seem to remember about the design if we have the time”
- The digital design must be well defined prior to attempting to capture it in HDL.
- The level of detail is debatable; however, if you cannot describe your design with timing diagrams, block diagrams, and some text, how can you expect to describe it with HDL accurately?

Process

Documentation

Peer reviews (desk check)

- A working review finds errors early in the design cycle; avoiding later changes that may compromise quality
- Requires enough staff and schedule for other qualified engineers to take time to review co-worker’s design

Change request log

- The quality of a good design that requires many changes is likely to suffer

Process
Design Capture

Capture HDL from a design document

- Design prior to coding vs. designing while coding
- Avoid the myth of "self-commenting code"; **the code tells me what it does, not what it is supposed to do.**

HDL Coding ~~Standard~~ Guidelines

- Enforce the use of HDL techniques that reduce the probability of error
- *INTUITIVE's* VHDL Capture Guideline is available free

Key point

HDLs and simulators make it easy to substitute action for thought; Design first, then code!

Process
Simulation

Functional simulation

- Proves that the HDL embodies the design correctly
- Absolutely necessary in every design

Lint warnings

- Warnings regarding legal, but inadvisable HDL syntax

Post synthesis, post place & route simulation

- Verifies that the synthesis tool did not unexpectedly add, remove, or change the implementation the designer described
- Recommended, but seldom available because the necessary models are unavailable

Process
Scripts

Simulation and synthesis should be run using (TCL) scripts

- The script captures the required steps to build (or rebuild) a project
- Eliminates errors due to forgetting build steps
- Frees the engineer to think about the design instead of focusing on the mundane tasks
- Facilitates future use of the design

Script Benefits

- Facilitates design understanding in the future...
- Frees the engineer from the equivalent of hourly typing quizes

Process
Revision Control

Importance of Revision Control

"If you're not using revision control, just stop developing; it's not worth your time." — Jack Ganssle

Revision Control Goals

- Should be able to reproduce the design
  - Design documents
  - Source HDL, constraints (pin locations, timing, etc)
  - Scripts
  - Test bench
- And reproduce the design environment
  - OS, OS patches
  - Tools, tool license

Management Experience

Understanding, Commitment, Staffing

- What is the level of management's understanding of FPGA technology?
- What is the level of commitment to:
  - An informed FPGA design process
  - Design team training
- Design team staffing
  - Understaffed teams skip process steps
  - This leads to avoidable errors

Independent Review

An IV&V team, at least as competent as the FPGA design team, should review the requirements, design document, HDL, simulation, etc.

Design Team Experience

FPGA design is digital circuit design

Writing reliable, synthesizable, efficient, synchronous HDL for FPGAs requires knowledge of digital design techniques

HDLs are not software programming languages

- Quality will suffer if the design team does not have a strong digital design background and/or relevant training.
- In HDL form or schematic form, digital design is still digital design
  - Interface to input and output circuits (including analog effects)
  - Timing margins
  - Synchronous vs. asynchronous
  - Metastability and re-clocking

First Order Quality Indicators

There are many quality indicators, these are the heavy hitters.

Design Techniques

- Reset source
- Reset synchronization
- Clock domains
- Synchronous design
- Finite state machines

Tool Usage

- Timing report
- Synthesis warnings

Design Techniques

Reset Synchronization

Reset source and synchronization

- Reset must be removed synchronously with the clock to avoid intermittent metastability failures
  - "Asynchronous & Synchronous Reset Design Techniques — Part Deux", Cummings et. al., Sunburst Design
  - Xilinx white paper 272

MAPLD 2004 (NASA Office of Logic Design)

"Unintended operation or lockup of finite state machines or systems may result if the flip-flops come out of reset during different clock periods. There is a potential for one or more uncontrolled metastable states. Therefore, only reset circuits that [attempt to] remove power on reset synchronously should be considered in hi-rel applications."

Design Techniques

Clock Domains

Digital Design Clocking Rules

1. Only use one clock!
2. When you need more than one clock, only use one clock!

Seriously...

FPGAs frequently have multiple clocks

- Each clock domain is synchronous to one clock
- Clock domain crossing is a critical design detail

Design Techniques

Synchronous design

FPGAs require synchronous design practices

- Asynchronous circuits will cause intermittent failures

Synchronous design rules

- All data are passed through combinatorial logic and flip-flops that are synchronized to a single clock.
- Delay is always controlled by flip-flops, not combinatorial logic.
- No signal that is generated by combinatorial logic can be fed back to the same group of combinatorial logic without first going through a synchronizing flip-flop.
- Clocks cannot be gated; clocks must go directly to the clock inputs of the flip-flops without going through any combinatorial logic.
- Do not clock entities or processes with outputs of other entities or processes.

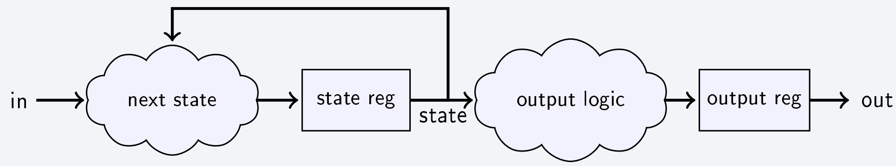
Ref: "Introduction to CPLD and FPGA Design", Bob Zeidman, Zeidman Consulting

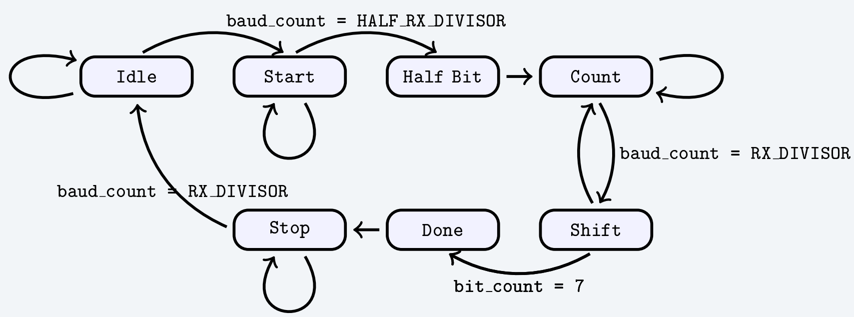
Design Techniques

State Machines

Disclaimer

A thorough discussion of Finite State Machines (FSM) is on the order of a semester long graduate level course.





Design Techniques

State Machines

Reasonable complexity of Conventional FSMs

- A good rule of thumb is no more than 20 “complex” states
- Break large state machines into several smaller ones
- Think about the complexity of the next state equation the tool will have to develop
- Think about your understanding of the design and the complexity of the implementation

HDL Capture of FSMs

Naive HDL for a 12-bit one-hot encoded FSM may have thousands of undefined states!

Timing Report

Critical Data

- The timing report is the vendor's estimate of how your design will perform in their part
- If they say you have little or no margin. . .

Use Vendor Tool Settings (Unless you have a PhD in semiconductor manufacturing)

- Vendor defaults for voltage, temperature have built-in margin
- Semiconductor physics can be non-linear
  - Changing the temperature from 85°C to 50°C due to your operating environment can mislead you. . .

Synthesis Tool Warnings

Intellectual Property (IP)

- Vendor provided IP notoriously produces many warnings
  - There may be a few critical warnings in thousands of mundane warnings
- Vendor IP can lead to several issues
  - License restrictions, obsolescence, support, etc.

No "Inferred Latch" warnings

- This is the tool telling you that **you have not completely defined your intent.**
- In order to provide the described behavior, the tool must insert a memory element (latch) where the designer did not request one.
- This is at the same level of concern as an IRS audit

Concluding Thoughts

Key Points

- FPGA design is digital logic circuit design
- The FPGA designer must follow a good design process to get good results
- The designer must be aware of how the HDL is physically implemented
- Functional simulation is required for every design
- There are many free resources and tutorials available

Further Reading I

C. Fulks and RC Cofer.
Best FPGA Development Practices.
Design West - ESC Summit; Class ESC-405, 2012.

NASA
Assurance of Complex Electronics.
http://www.hq.nasa.gov/office/codeq/software/ComplexElectronics/.
NASA 2009.

Charles Fulks.
VHDL Capture Guidelines.
Intuitive Research and Technology Corporation, 2014

P. Ashenden.
The Designer's Guide to VHDL.
Morgan Kaufmann Pub, 2008.

INTUITIVE Research and Technology www.irtc-hq.com