Introduction to Digital Signal Processing on FPGA

Qiang Du

USPAS Houston, January 25, 2023

Lawrence Berkeley National Lab

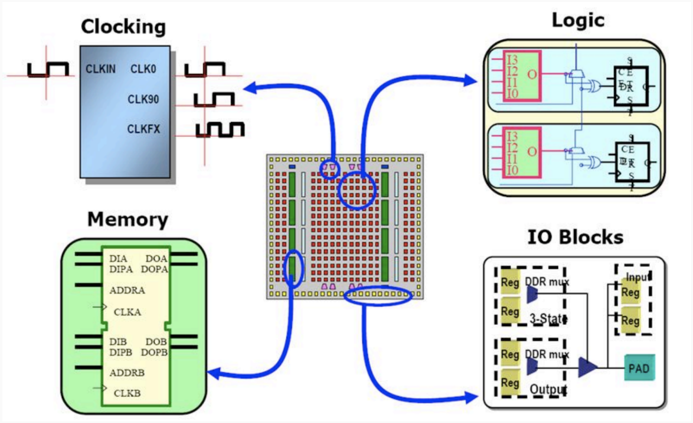
Table of contents

Introduction to digital circuit design

FPGA: Field Programmable Gate Array

Typical elements:

- Random Access Memory (RAM)
- Logic gates
- DSP
- Input / Outputs
- Clock management units
- Phase Locked Loop (PLL)
- Multi-Gigabit Transceivers
- ...



How to design a digital circuit

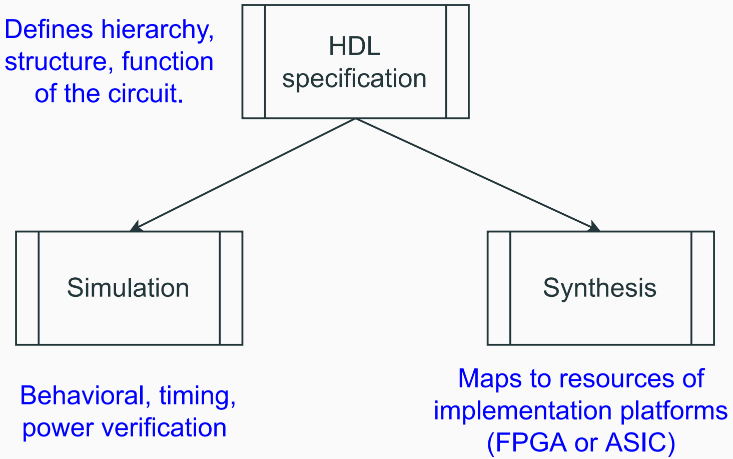
Schematic Circuit Design

- Advantage
  - Block diagrams, imply physical hardware
  - Intuitive, match gate-level logic
- Disadvantage
  - Requires graphical editor tool
  - Hard to track changes
  - Scalability suffers when hierarchy grows

Hardware Description language (HDL)

- Advantage
  - Text based, no special tool needed
  - Version control friendly
  - Scalable (similar to Object Oriented Programming)
- Disadvantage
  - Not so intuitive
  - Not directly showing the hardware primitives

Digital circuit design methodology


Defines hierarchy, structure, function of the circuit.
HDL specification
Simulation
Synthesis
Behavioral, timing, power verification
Maps to resources of implementation platforms (FPGA or ASIC)

Lab 3: LED blinker, Lab 4: Frequency counter

Hardware Description Languages

	Strong typing	Style	typedef	Structures	Class	Interface
VHDL	Yes	ADA	Yes	Yes	No	Partial
Verilog	No	C	No	No	No	No
SystemVerilog	Partial	C	Yes	Yes	Yes	Yes

VHDL	More syntactic overhead, semantically close to Verilog
Verilog	Mature set of commercial / open-source tools
SystemVerilog	Enhancement to Verilog, less-mature tools

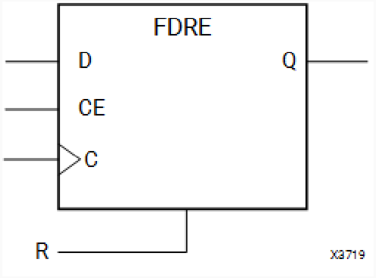
Verilog modules and instantiation: combinational and sequential logic

'led_test.v'

```
module led_test #(                                  // module define
    parameter MSB = 27 )(                           // parameter define
    input clk,
    input reset,                                    // port define
    output [3:0] led) ;                             // multi-bit signal
reg [31:0] cnt=0;                                   // register that holds states
always @(posedge clk) begin                         // sensitivity list, non-blocking
    cnt <= reset ? 32'h0 : cnt + 1'b1;              // constant, mux,
end                                                 //   assignment (defer), sequential
assign led = cnt[MSB:MSB-3];                        // blocking assignment (no defer)
endmodule                                           //   combinational
```

Use Non-blocking for sequential logic. Use blocking for combinatorial logic. details

D Flip-flop (FF)

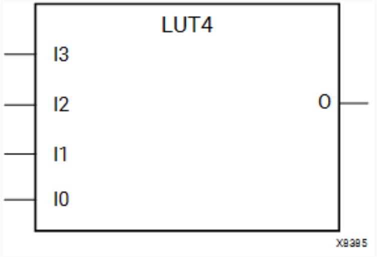


Xilinx Primitive: D Flip-Flop with Clock Enable and Synchronous Reset, FDRE

Verilog example

```
module dff(
    input D,
    input clk,
    input reset,
    output reg Q
);
    always @(posedge clk) begin
        if (reset) // synchronous reset
            Q <= 1'b0;
        else
            Q <= D;
    end
endmodule
```

Look-up Table (LUT)



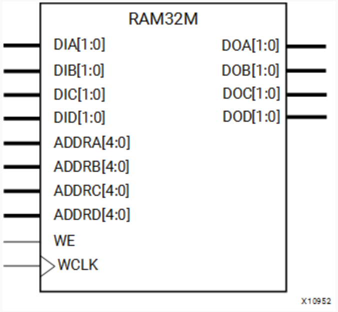
Xilinx Primitive: 4-Bit Look-Up Table, LUT4

Verilog example: Combinational Logic

```
module lut4 (
    input [3:0] I,
    output O
);
    assign O = I[0] | (&I[3:1]);
endmodule
```

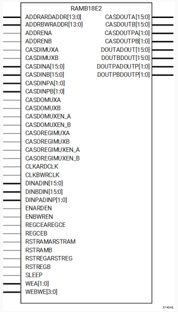
Internal Random Access Ram (RAM)

Distributed RAM: Tiny and flexible, but not efficient in terms of area.



Xilinx Primitive: 256-Deep by 1-Wide Random Access Memory (Select RAM), RAM256X1D

Block RAM: Big RAM blocks located in dedicated areas, limited numbers;



Xilinx Primitive: 18K-bit Configurable Synchronous Block RAM, RAMB18E2

Internal Random Access Ram (RAM)

- First In, First Out buffer (FIFO)
- Single, Dual, Multiple port Memory
- 1:N or N:1 aspect ratio
- Clock domain crossing

Verilog example: Dual port RAM

```
module dpram #(
    parameter AW = 8,
    parameter DW = 8
) (
    input clka, clkb, wena,
    input [AW-1:0] addra, addrb,
    input [DW-1:0] dina,
    output [DW-1:0] douta, doutb
);
parameter SIZE = (32'b1<<AW)-1;
reg [DW-1:0] mem[SIZE:0];
reg [AW-1:0] ala=0, alb=0;

assign douta = mem[ala];
assign doutb = mem[alb];
always @(posedge clka) begin
        ala <= addra;
        if (wena) mem[addra]<=dina;
end
always @(posedge clkb) begin
        alb <= addrb;
end
endmodule
```

How to do 4-bit adding using logic gates?

Do it by hand:

	a3	a2	a1	a0
+	b3	b2	b1	b0
c	r3	r2	r1	r0

Add a0 and b0:

a0	b0	r0	c0
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$r = a \oplus b, \qquad c = a \& b$$

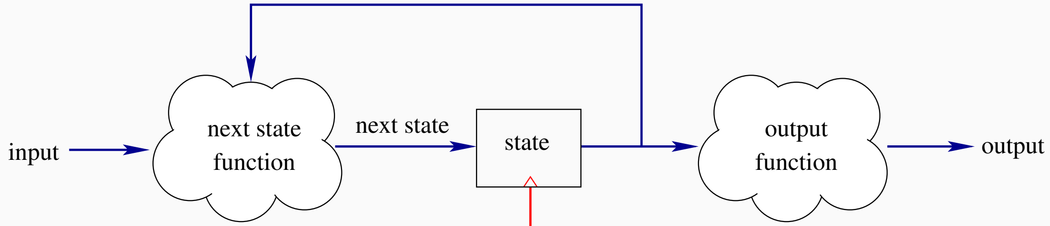
Add a1 and b1:

c0	a1	b1	r1	c1
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

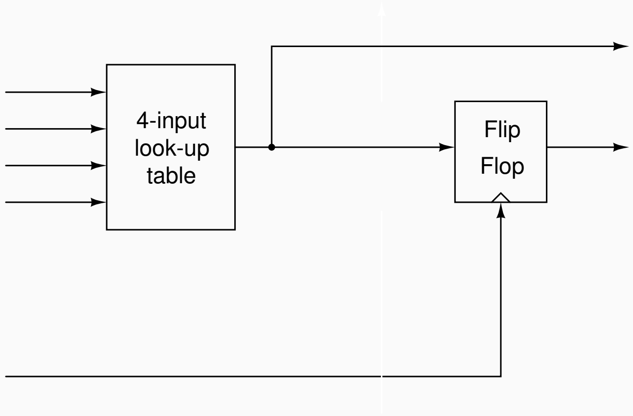
$$r = a \oplus b \oplus c, \quad c_o = a\&b + a\&c_i + b\&c_i$$

Programmable Logic Cells

Logic Cell

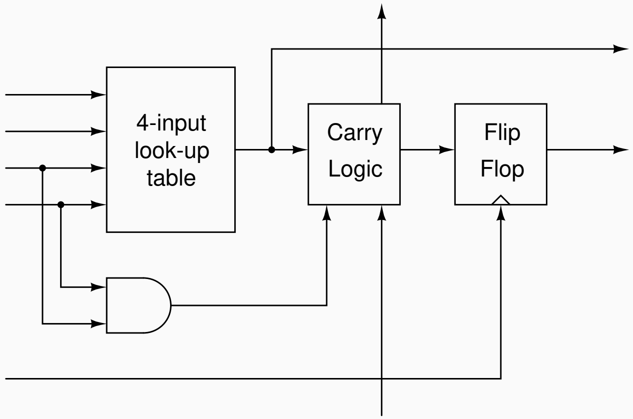

input
next state function
next state
state
output function
output
clock

Logic Cell: Simplist concept



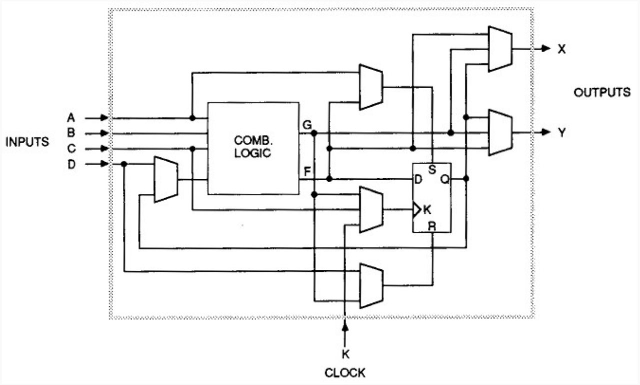
And, or, xor, multiplex, two-way mux with force-to-constant

Logic Cell: Simplist concept with a carry chain

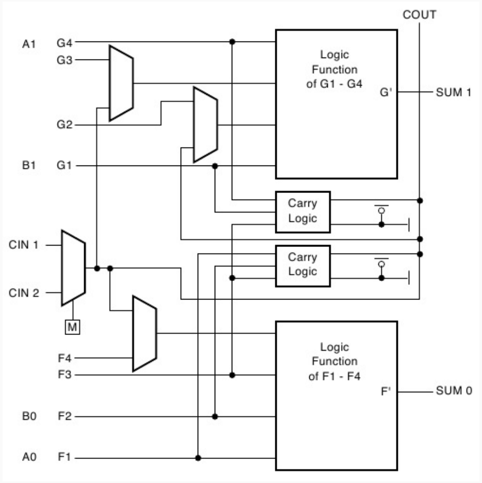


Add, subtract, gated-add, add/sub

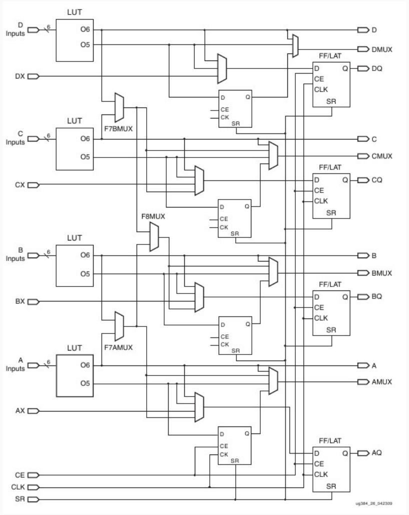
History: First commercial FPGA (XC2064) in 1985 had 64 of these cells



History: Modern carry chain (XC4000) from 1991



History: Modern carry chain (Spartan6) from 2009



Datasheet: Xilinx 7 series family, September 2020

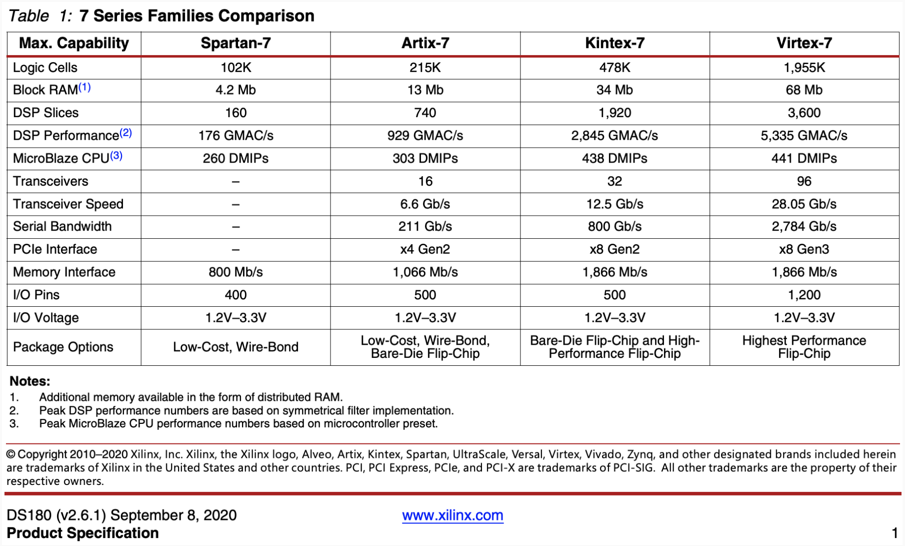


Table 1: **7 Series Families Comparison**

Max. Capability	Spartan-7	Artix-7	Kintex-7	Virtex-7
Logic Cells	102K	215K	478K	1,955K
Block RAM[1]	4.2 Mb	13 Mb	34 Mb	68 Mb
DSP Slices	160	740	1,920	3,600
DSP Performance[2]	176 GMAC/s	929 GMAC/s	2,845 GMAC/s	5,335 GMAC/s
MicroBlaze CPU[3]	260 DMIPs	303 DMIPs	438 DMIPs	441 DMIPs
Transceivers	–	16	32	96
Transceiver Speed	–	6.6 Gb/s	12.5 Gb/s	28.05 Gb/s
Serial Bandwidth	–	211 Gb/s	800 Gb/s	2,784 Gb/s
PCIe Interface	–	x4 Gen2	x8 Gen2	x8 Gen3
Memory Interface	800 Mb/s	1,066 Mb/s	1,866 Mb/s	1,866 Mb/s
I/O Pins	400	500	500	1,200
I/O Voltage	1.2V–3.3V	1.2V–3.3V	1.2V–3.3V	1.2V–3.3V
Package Options	Low-Cost, Wire-Bond	Low-Cost, Wire-Bond, Bare-Die Flip-Chip	Bare-Die Flip-Chip and High-Performance Flip-Chip	Highest Performance Flip-Chip

Notes:

1. Additional memory available in the form of distributed RAM.
2. Peak DSP performance numbers are based on symmetrical filter implementation.
3. Peak MicroBlaze CPU performance numbers based on microcontroller preset.

Datasheet: Xilinx Ultrascale series family, December 2022

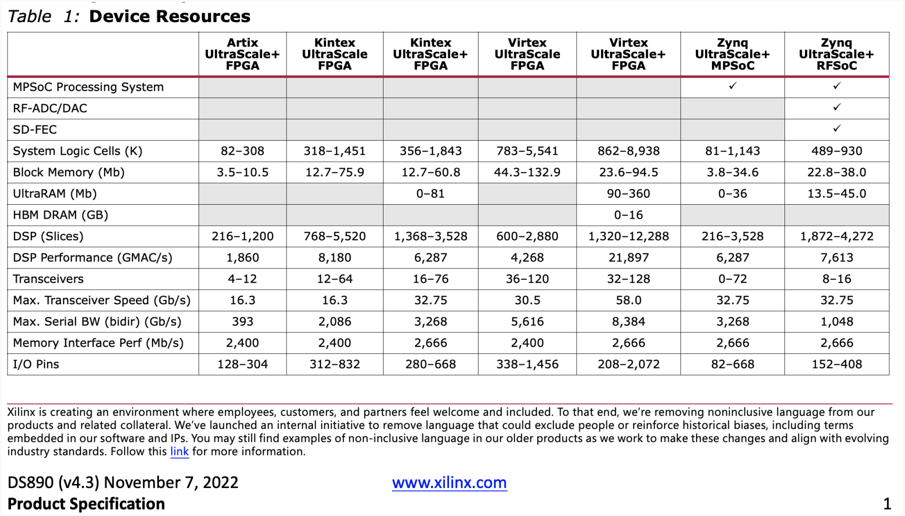


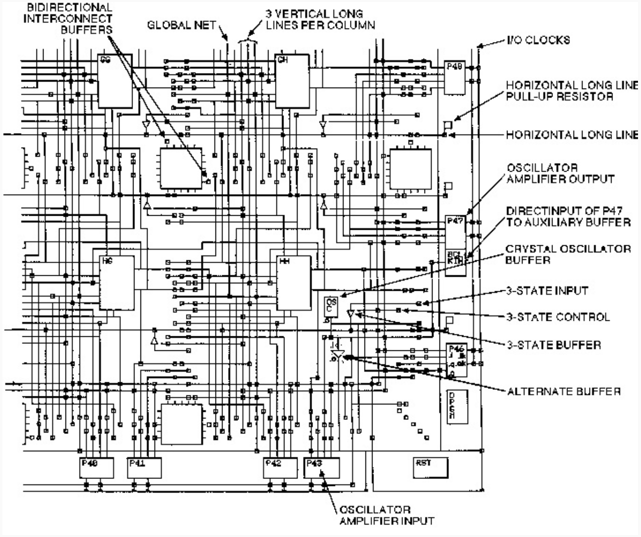
Table 1: **Device Resources**

	Artix UltraScale+ FPGA	Kintex UltraScale FPGA	Kintex UltraScale+ FPGA	Virtex UltraScale FPGA	Virtex UltraScale+ FPGA	Zynq UltraScale+ MPSoC	Zynq UltraScale+ RFSoC
MPSoC Processing System						✓	✓
RF-ADC/DAC							✓
SD-FEC							✓
System Logic Cells (K)	82–308	318–1,451	356–1,843	783–5,541	862–8,938	81–1,143	489–930
Block Memory (Mb)	3.5–10.5	12.7–75.9	12.7–60.8	44.3–132.9	23.6–94.5	3.8–34.6	22.8–38.0
UltraRAM (Mb)			0–81		90–360	0–36	13.5–45.0
HBM DRAM (GB)					0–16		
DSP (Slices)	216–1,200	768–5,520	1,368–3,528	600–2,880	1,320–12,288	216–3,528	1,872–4,272
DSP Performance (GMAC/s)	1,860	8,180	6,287	4,268	21,897	6,287	7,613
Transceivers	4–12	12–64	16–76	36–120	32–128	0–72	8–16
Max. Transceiver Speed (Gb/s)	16.3	16.3	32.75	30.5	58.0	32.75	32.75
Max. Serial BW (bidir) (Gb/s)	393	2,086	3,268	5,616	8,384	3,268	1,048
Memory Interface Perf (Mb/s)	2,400	2,400	2,666	2,400	2,666	2,666	2,666
I/O Pins	128–304	312–832	280–668	338–1,456	208–2,072	82–668	152–408

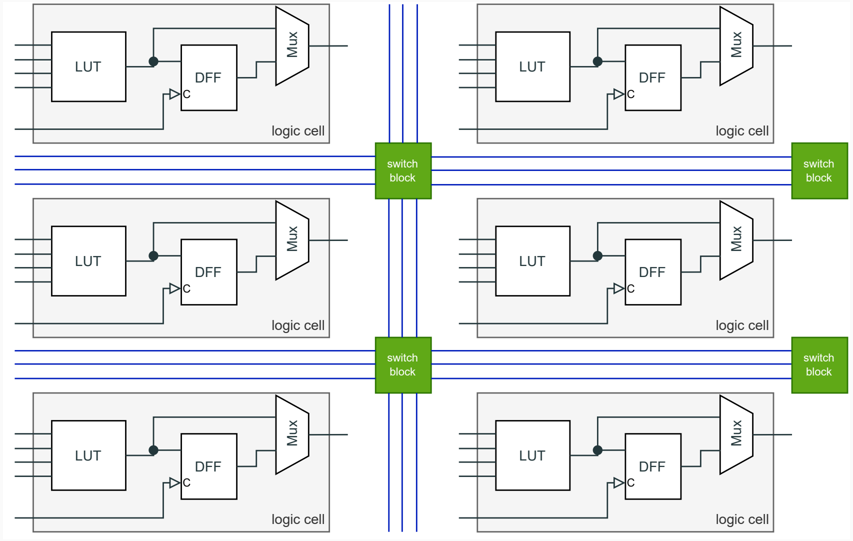
Routing

Routing

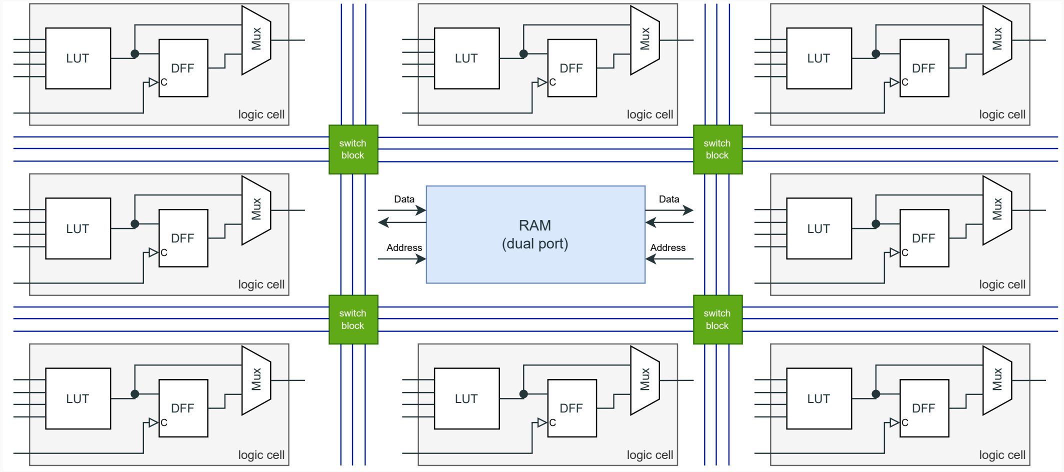
Well documented routing from 1998 (XC3000).
Routing needs scale super-linearly with chip size; routing hardware is now much more capable, complex, and less well documented.


BIDIRECTIONAL INTERCONNECT BUFFERS
GLOBAL NET
3 VERTICAL LONG LINES PER COLUMN
I/O CLOCKS
HORIZONTAL LONG LINE PULL-UP RESISTOR
HORIZONTAL LONG LINE
OSCILLATOR AMPLIFIER OUTPUT
DIRECT INPUT OF P47 TO AUXILIARY BUFFER
CRYSTAL OSCILLATOR BUFFER
3-STATE INPUT
3-STATE CONTROL
3-STATE BUFFER
ALTERNATE BUFFER
OSCILLATOR AMPLIFIER INPUT
GG
CH
HG
HH
P48
P47
P46
P40
P41
P42
P43
RST
OSC

Programmable Logic: Interconnect Logic Cells



Programmable Logic: Interconnect Logic Cells with RAM



Clocks

Clocks

- Handful of well-defined domains: good
- Clock-enable on registers: good
- Low latency data path involving multiple clock domains: bad
- Gated clocks: terrible

As FPGAs get larger and faster, demands on the clock tree get harsher. Newer chips deprecate global clocks; clock-domain crossing is needed between zones. Think of it as several chips in one package (sometimes literally true). Tools and design practices must be adjusted to cope.

Clocks

Generally our designs have at least two clock domains:

- High throughput DSP has no wiggle-room on its speed, has to match ADC
- Communications (e.g., Ethernet, USB, VME, PCI)

Many unavoidable reasons to increase from the minimum, e.g., White Rabbit leaf node needs one or two more domains to function correctly.

If you have lower throughput DSP stuff to do, *and* wish to re-use someone else's "IP" (e.g., a soft core) that may not make timing with your main DSP clock, you may need to generate a subharmonic clock for that sub-circuit. That will count as a new clock domain! Example:

- Z80 from OpenCores at 40 MHz
- LM32 at 80 MHz

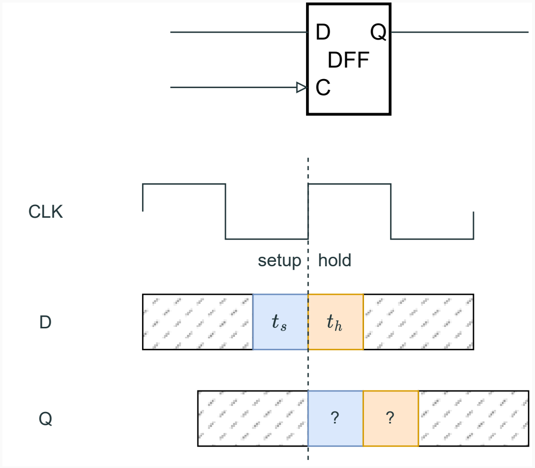
Clocks

Only a finite number of clock domains are supported on any given FPGA!

Miscellaneous “low speed” logic, e.g., a serial DAC or thermometer, doesn’t need a separate clock domain! If the logic is clean, it will make timing at the primary clock rate. Simply put clock-enables on all its registers to slow down its sequencing.

- Saves the clock domains for essential uses
- Simplifies attachment to the rest of the chip’s logic.

Metastability

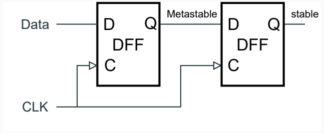


- A metastable state is one in which the output of Flip-Flop is unknown,
- When setup or hold time are violated.
- Input data must be stable within t_s and t_h.
- Signal propagation time must be analyzed to avoid metastability
- Most conditions occur in:
    - Sampling a signal external to the FPGA (such as ADC, clock);
    - Crossing clock domains;

Crossing clock domains

Multi-cycle signal

problem	Metastability
solution	Pipeline twice in receiver domain.



Single cycle flag

problem	Missing flag, metastability
solution	Convert flag to toggle, latch twice

Data bus (≥ 2 bits)

problem	misaligned bits
solution1	Gray code
solution2	First-In-First-Out (FIFO) buffer

Memory Hierarchy

On-chip memory

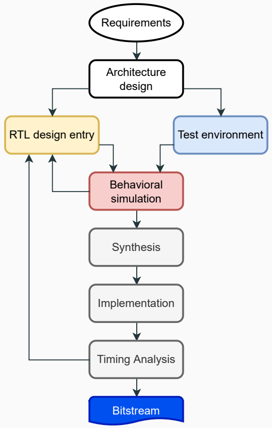
- Small, granular, flexible, fast:
  - Xilinx: 16 to 64 bits each, one-bit access
  - Altera: 32x20 or 64x10 in some chip families
- Block memory: dual-port, 4K to 36K bits each, 1 to 36 bit data ports
- Large block: single-port (rare)

On-chip memory: just inferred from HDL

Off-chip memory: slow, more difficult to use and simulate

Putting things together

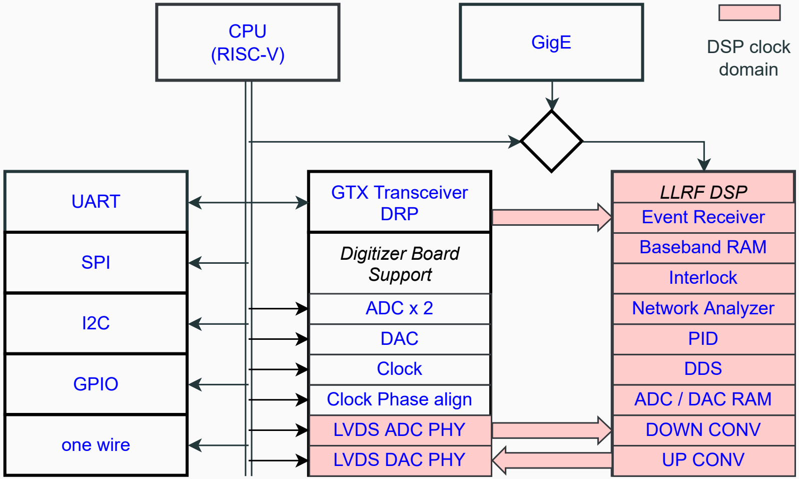
Design Process:



- Hardware Description Language (HDL):
  - VHDL, Verilog, SystemVerilog
- Simulation tools:
  - Icarus Verilog
  - Verilator
  - Mentor Graphics ModelSim, Xilinx Vivado Simulator, etc.
- Synthesis tools:
  - AMD Xilinx Vivado
  - Intel Quartus
  - Yosys Open Synthesis Suite

Check your design (timing, utilization, power) reports!

LLRF system architecture: System-On-Chip


CPU (RISC-V)
GigE
DSP clock domain
UART
SPI
I2C
GPIO
one wire
GTX Transceiver DRP
Digitizer Board Support
ADC x 2
DAC
Clock
Clock Phase align
LVDS ADC PHY
LVDS DAC PHY
LLRF DSP
Event Receiver
Baseband RAM
Interlock
Network Analyzer
PID
DDS
ADC / DAC RAM
DOWN CONV
UP CONV