An Introduction to FPGAs

Thierry Moreau
CSE548 17sp

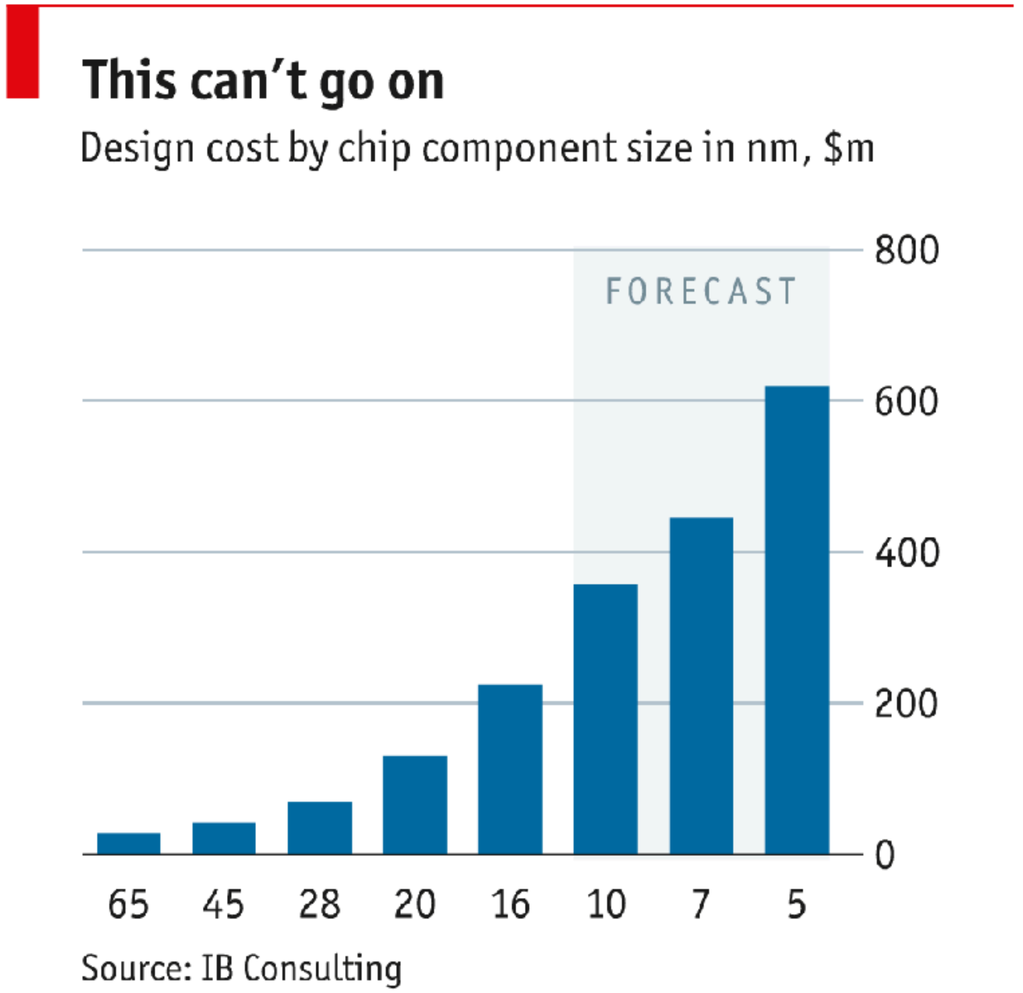
Talk Overview

- **What are FPGAs?**
- How do you program FPGAs?
- What can you do with FPGAs?

Limitations of ASICs

Tape-out costs for ASICs is exorbitant
10x cost gap between 16nm and 65nm

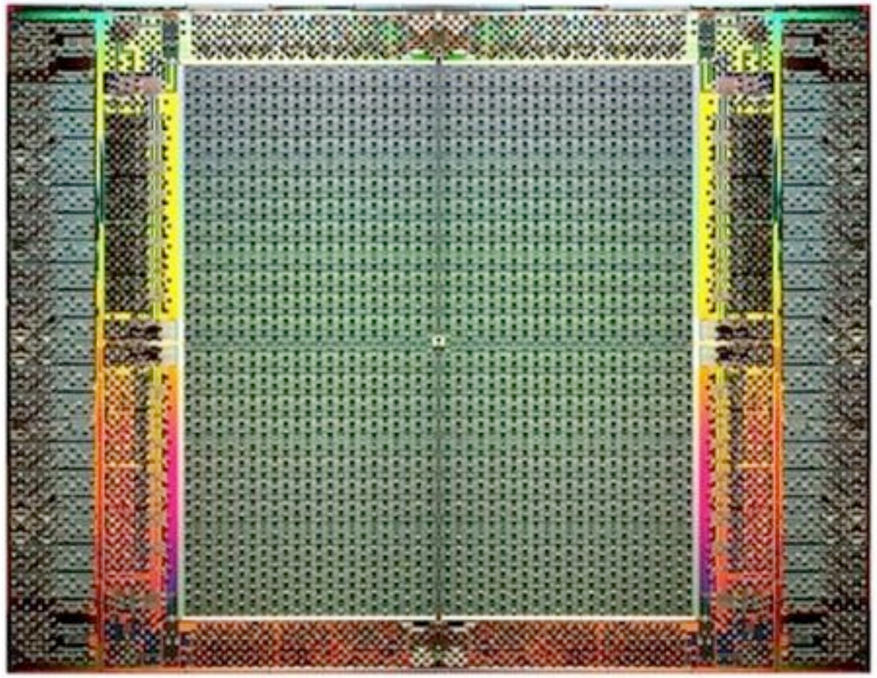
Risky bet to design hardware accelerators for ever-changing applications (think ML, search etc.)



FPGA (Field Programmable Gate-Arrays)

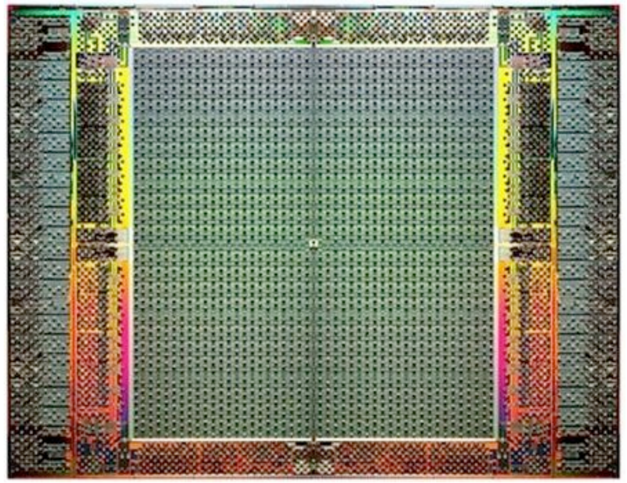
FPGAs provide a sea of programmable logic components that can be fully customizable.

Think of it as a virtual layer to implement any ASIC design on top of a fixed ASIC design.

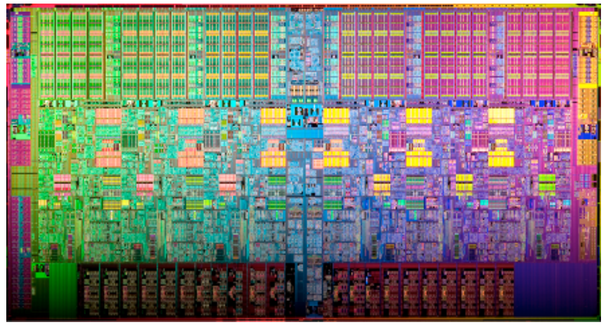


Stratix IV die photo

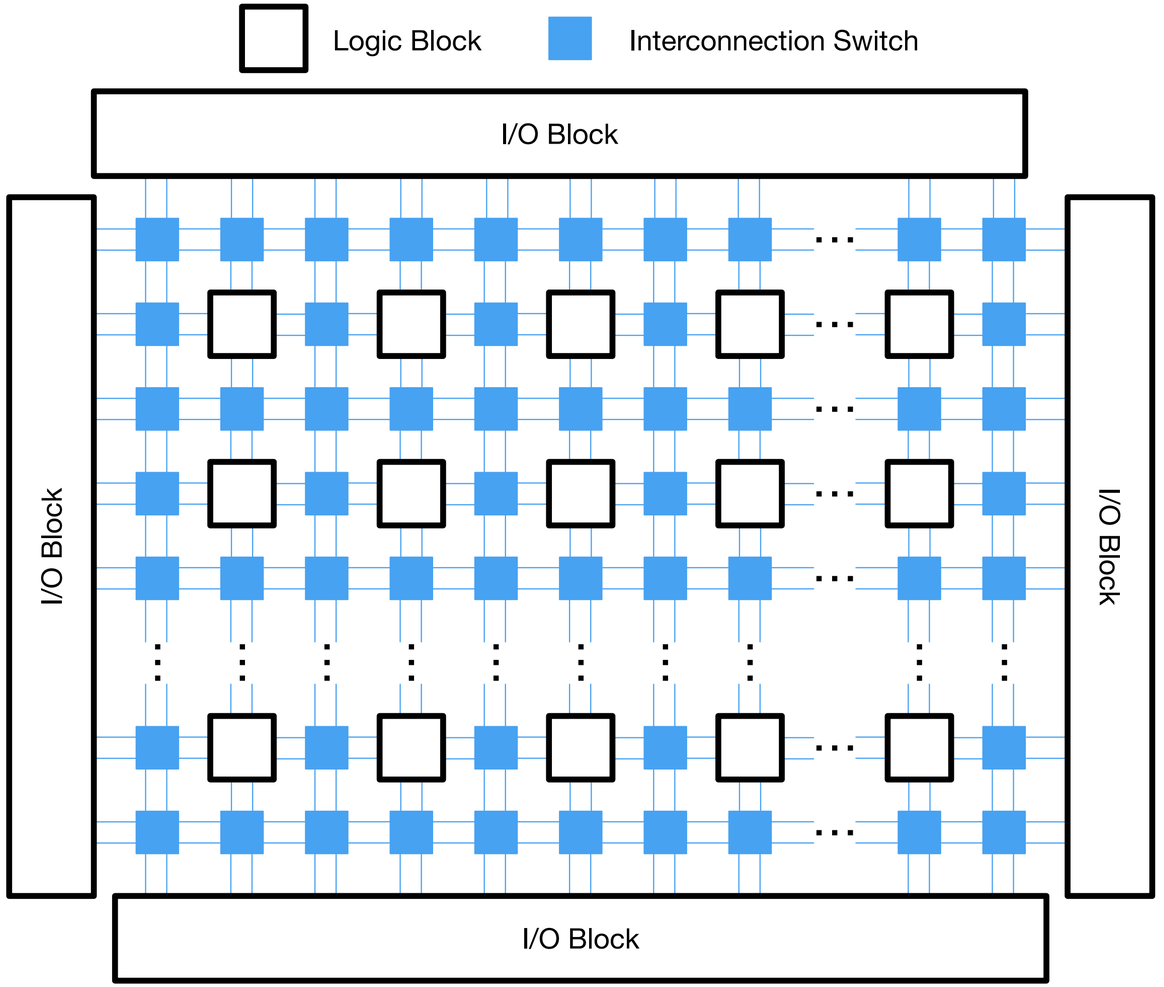
FPGA Architecture - High Level Structure



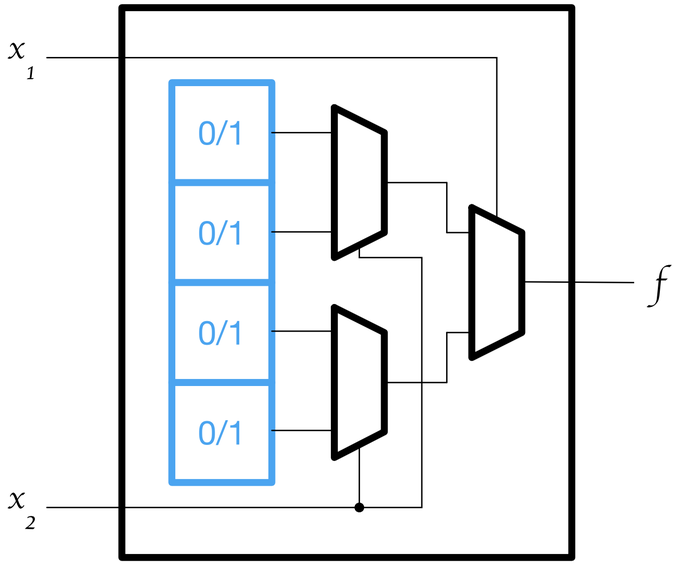
Stratix IV



Westmere Xeon for comparison



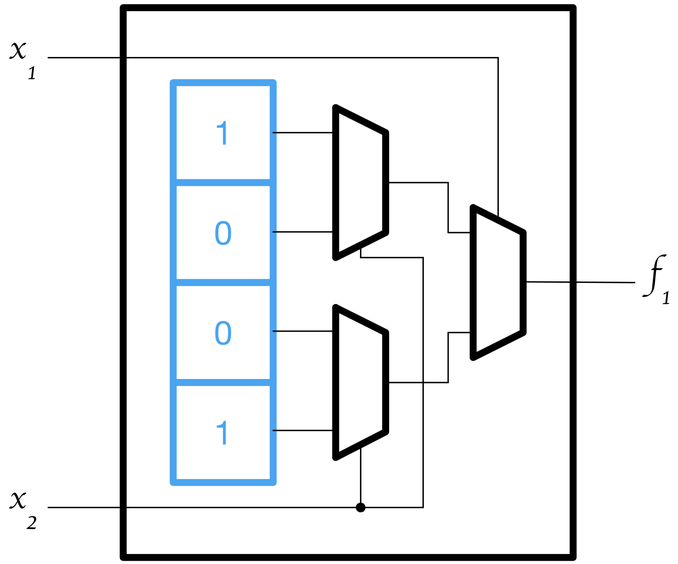
FPGA Architecture - Programmable LUTs



(a) Circuit for a two-input LUT

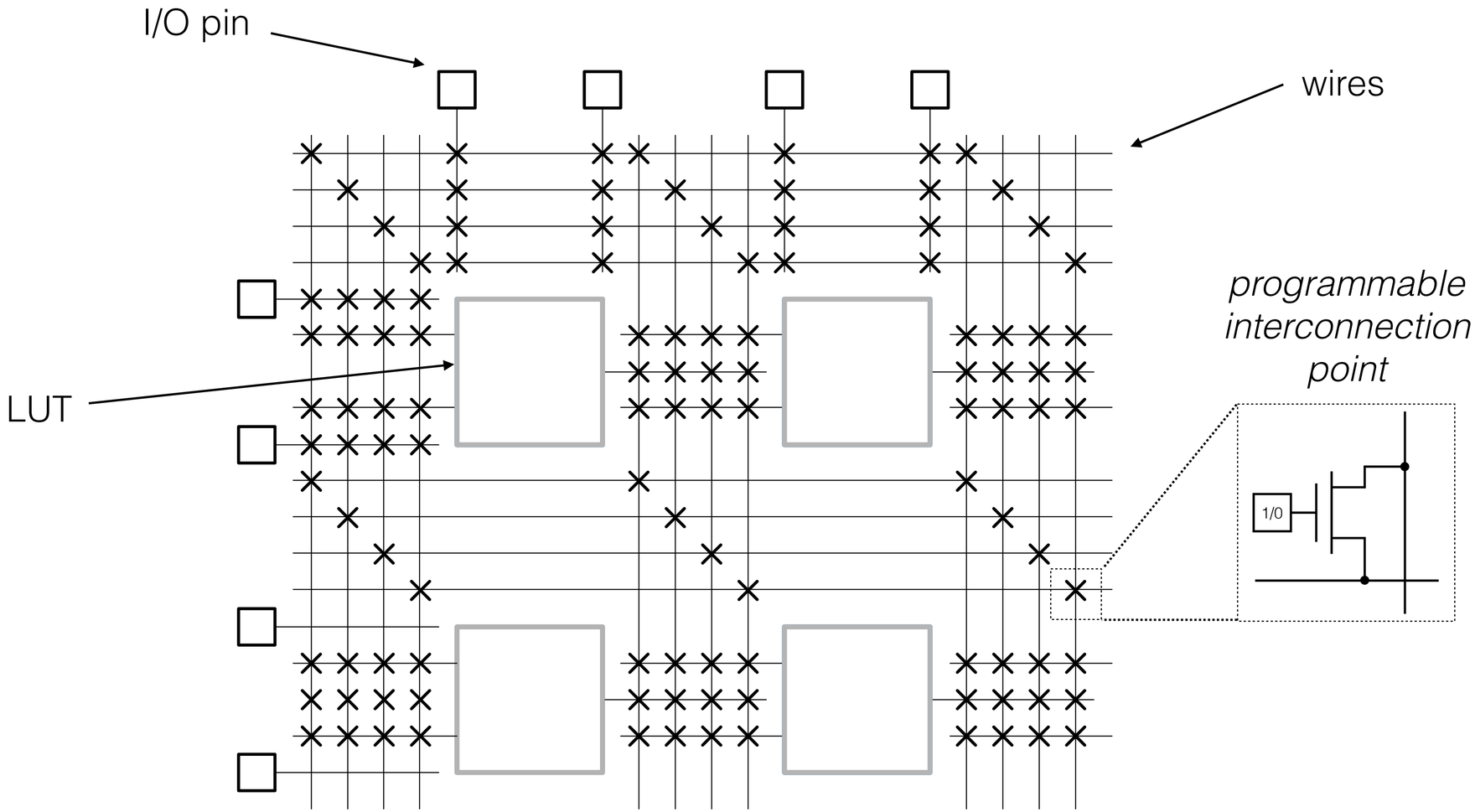
x_1	x_2	f_1
0	0	1
0	1	0
1	0	0
1	1	1

(b) $f_1 = \bar{x}_1 \bar{x}_2 + x_1 x_2$



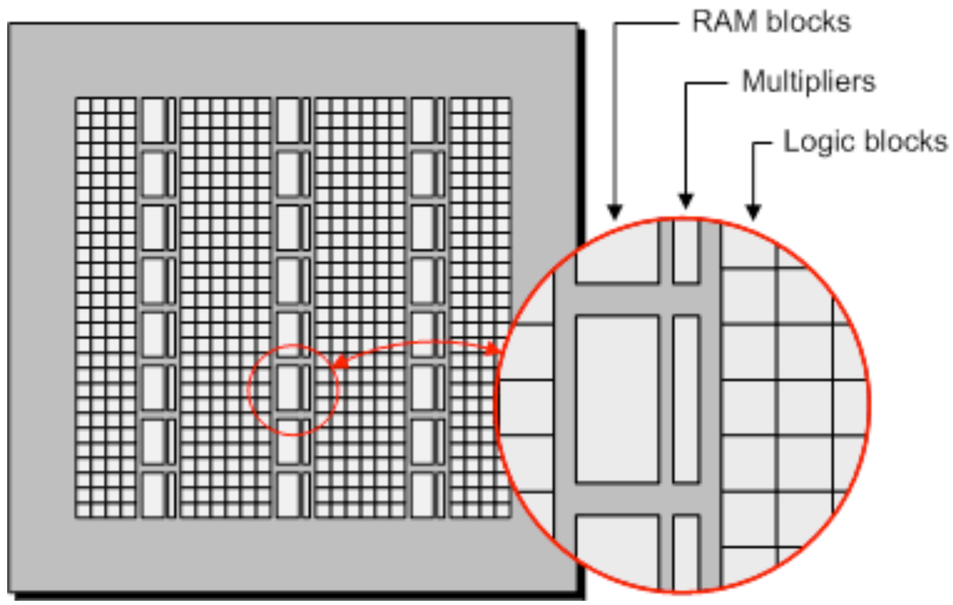
(c) Two-input LUT programmed to implement f_1

FPGA Architecture - Interconnection Fabric

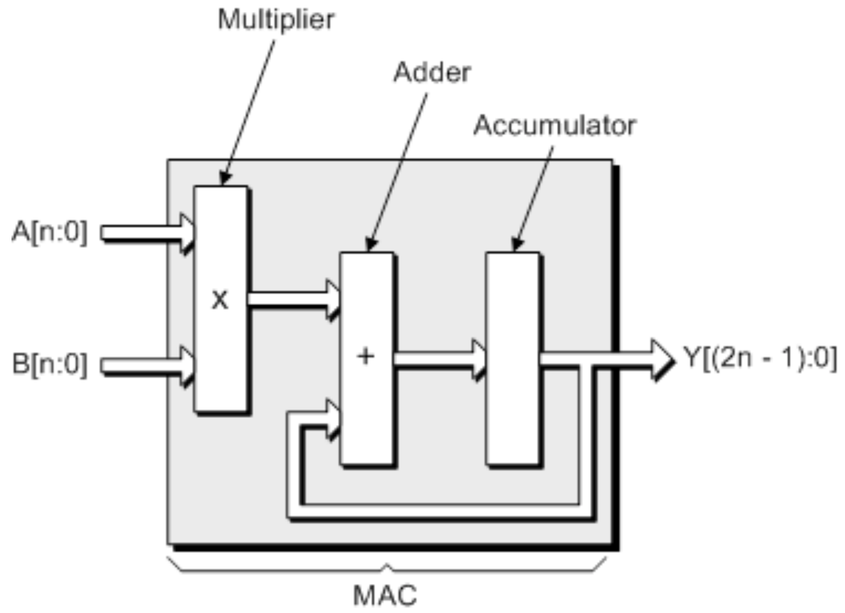


FPGA Architecture - Hard Components

Certain functions are commonly used across many designs:
thus it's worth *hardening* those modules

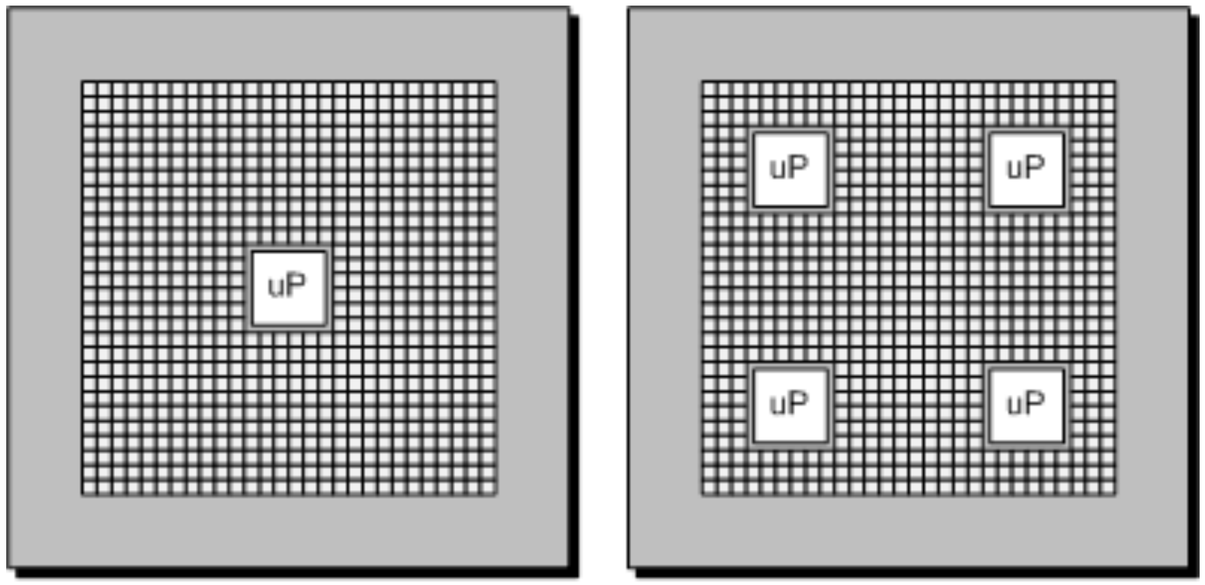


FPGA Architecture - DSP blocks



Digital Signal Processing (DSP) blocks provide dense fixed-point compute capabilities.

FPGA Architecture - Embedded Processors



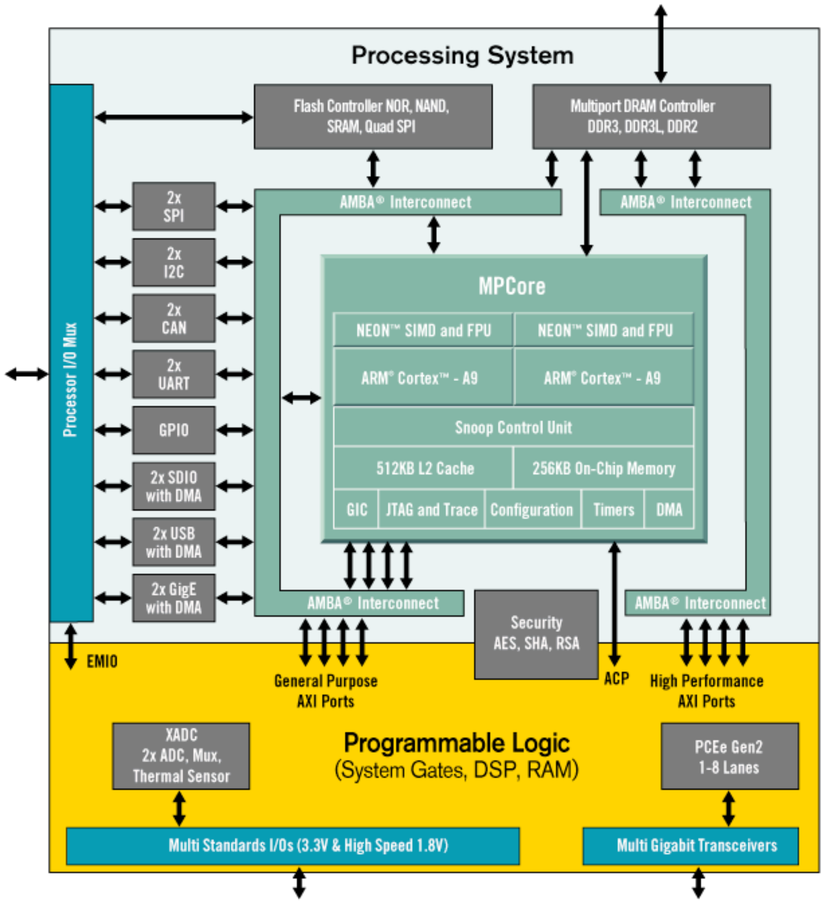
(a) One embedded core (b) Four embedded cores

Sometimes a small micro-processor can help offload simple book keeping tasks.

FPGA example: Zynq Programmable SoC

General ARM Cortex-A9 Processor

Low-Power FPGA fabric that shares the memory bus with the CPU



What is the FPGA-ASIC gap?

What is the cost of reprogrammability?

~40x area overhead based on LUT-only design*
~21x area overhead with of DSPs and Memory*

Corresponds to the area overhead of going from 14nm to 65nm!
Given that a 65nm tapeout it still expensive and time consuming, it is worth going to FPGA for many custom designs!

* Kuon et al. Measuring the Gap between FPGAs and ASICs. [FPGA06]

Talk Overview

- What are FPGAs?
- **How do you program FPGAs?**
- What can you do with FPGAs?

FPGA Programming Basics

Verilog - funky hardware description language

Logic in Verilog

Module definition of two identical **`and`** gates.

One gate declaration uses a continuous assignment with **`assign`**, while the other uses procedural assignment with **`always`**.

```
module foo (a,b,f,g);
    input wire a, b;
    output wire f;
    output reg g;

    assign f = a && b;
    always @(*)
        g = a && b;
endmodule
```

Hardware Registers in Verilog

Registers are used to hold state.

At every positive edge of the clock, the register captures the input value and holds it until the next positive edge of the clock.

Don’t confuse registers in Verilog with actual hardware registers!

```
input wire nextFoo;
reg foo;

always @(posedge clk)
    foo <= nextFoo;
```

Good crash-course in Verilog from MIT here and here

High-Level Synthesis

Verilog is an ugly language and is difficult to test and debug.

Is there a more approachable design entry language for software programmers?

Yes! It's called high-level synthesis and the tools are maturing.
Intel (formerly Altera) offers an OpenCL compiler, and Xilinx offers a C compiler.

You'll learn how to use HLS in your homework!

Altera® SDK for OpenCL™

FPGA Compilation: Overview

OpenCL/C file

High Level Synthesis

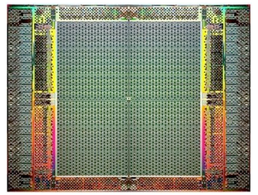
Verilog file

Logic Synthesis

Placement and Route

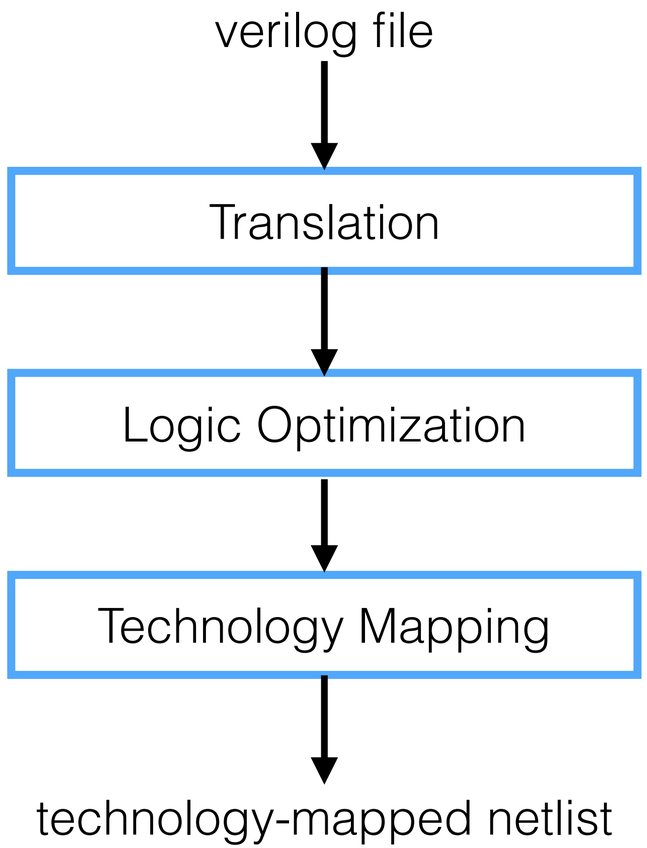
Bitstream Generation

bit stream



How do you go from a hardware specification in Verilog to a bit stream that configures the FPGA to implement the desired hardware circuit?

FPGA Compilation: Logic Synthesis



Translation:

- Convert HDL to boolean equations and latches

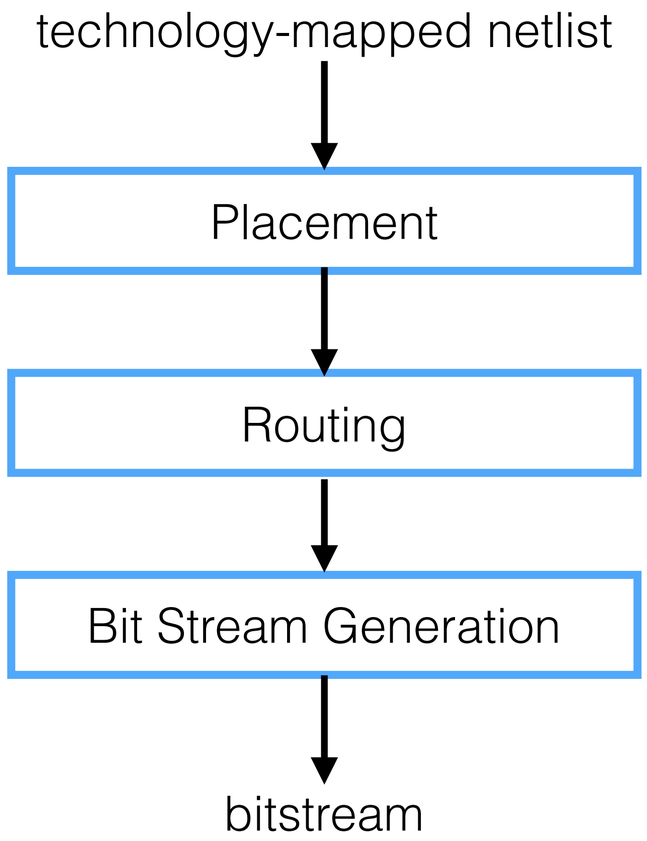
Logic Optimization:

- Reduce area and delay of circuit performing standard boolean optimization

Technology Mapping:

- Implement the equations and latches may mapping those to LUTs and other components

FPGA Compilation: Place & Route



Placement:

- Pin & gate assignment
- Optimizing position and orientation of cells
- Objective: minimize area (NP-complete)

Routing:

- Connect cells pins with wires
- Objective: minimize delay, area (NP-complete)

Bit-stream generation:

- Produces the bits that will configure the FPGA to do what we want

FPGA Compilation: Example

Step 1: Logic Synthesis for a 2-input LUT-based FPGA

```
module foo (a,b,f);
    input wire a, b, c;
    output wire f;

    assign f = a and b or (~b and c);
endmodule
```


$$f(x_1, x_2, x_3) = x_1 x_2 + \neg x_2 x_3$$

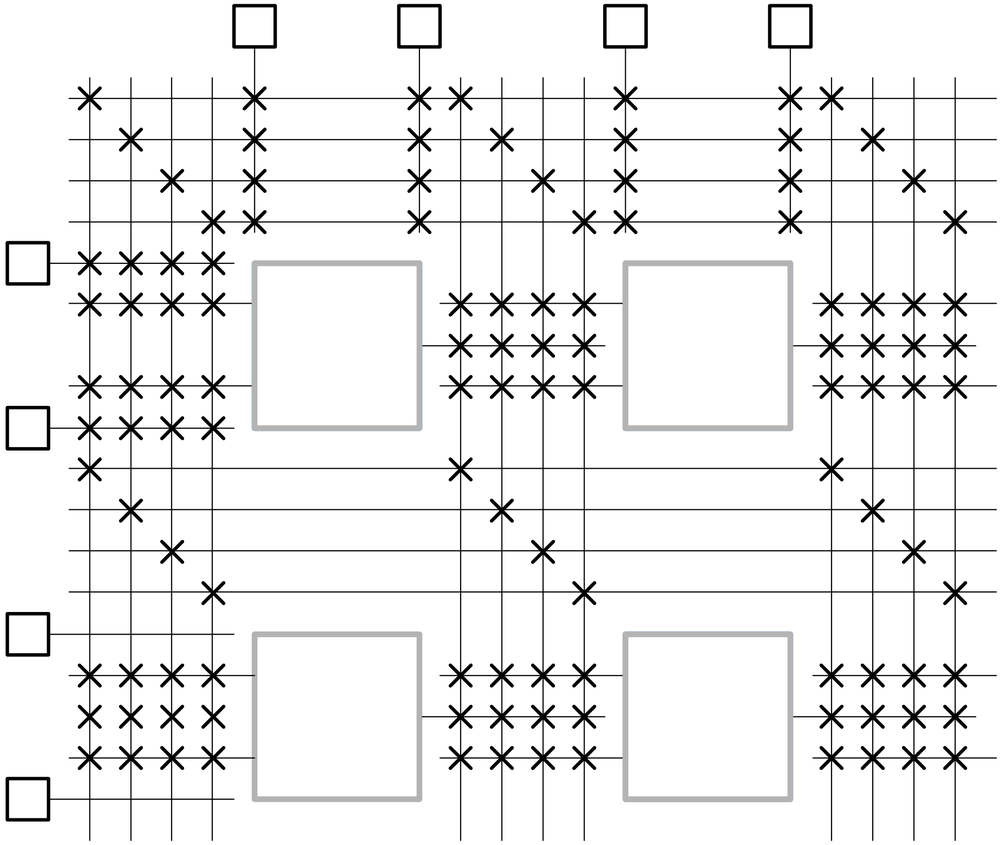
$$f_1 = x_1 x_2$$

$$f_2 = \neg x_2 x_3$$

$$f = f_1 + f_2$$

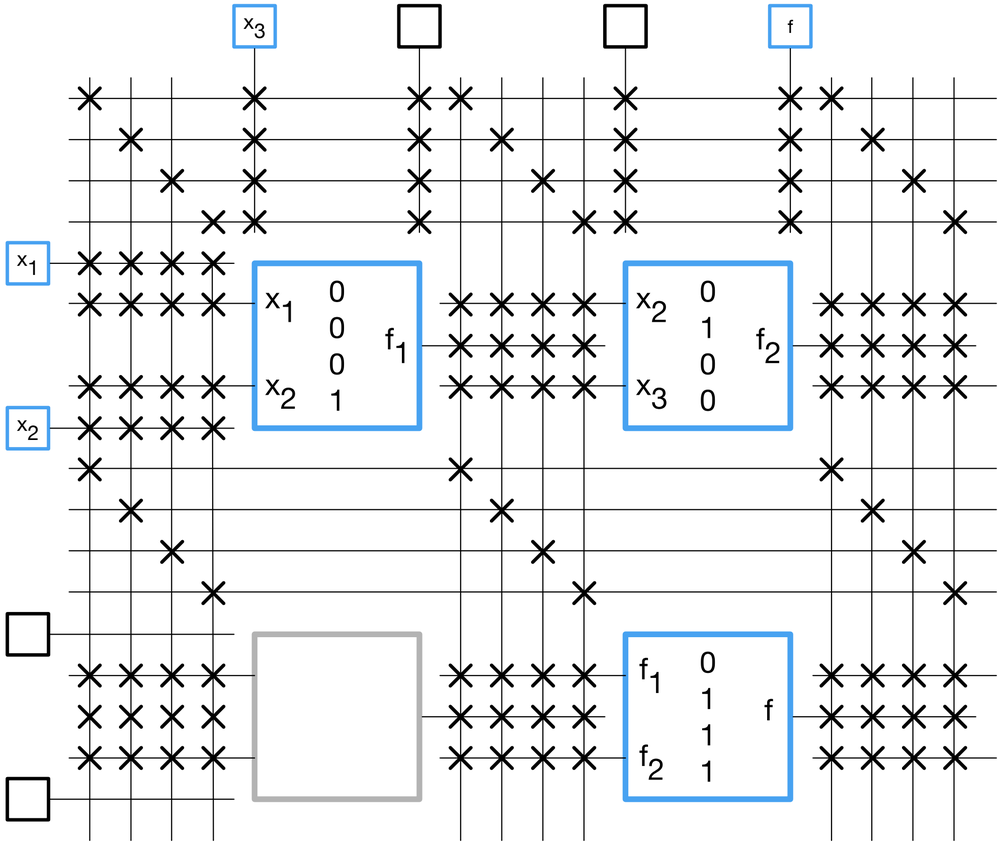
FPGA Compilation: Example

Step 2: Placement



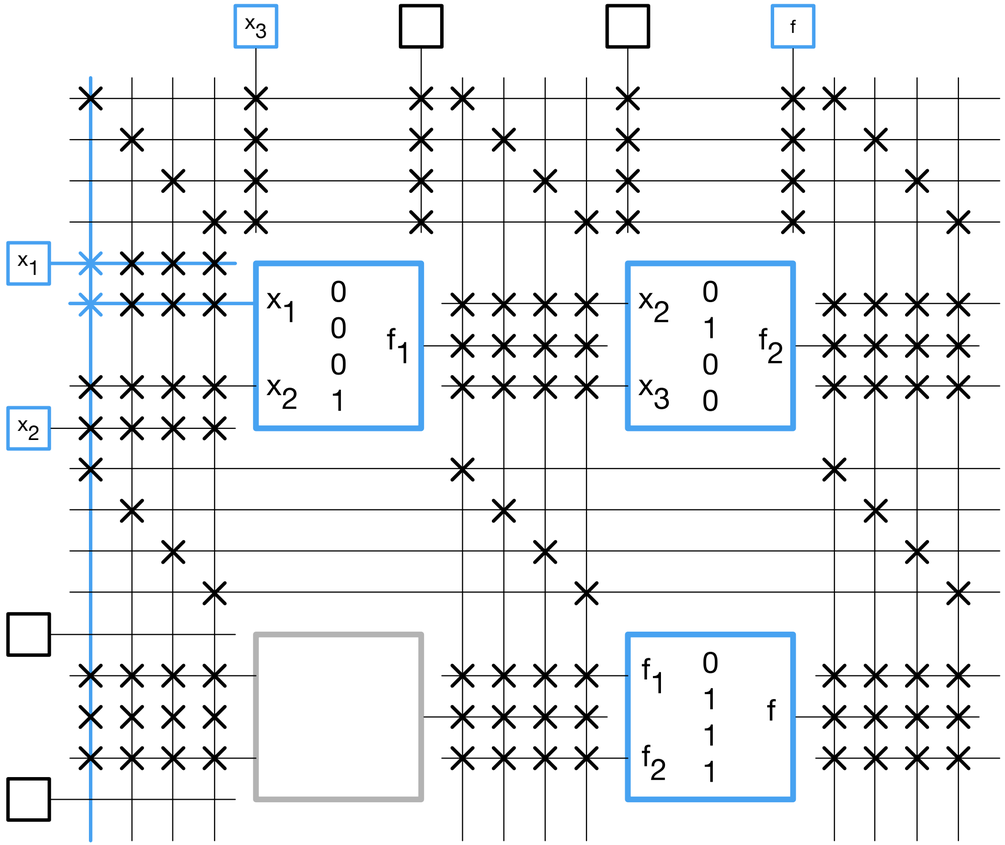
FPGA Compilation: Example

Step 2: Placement



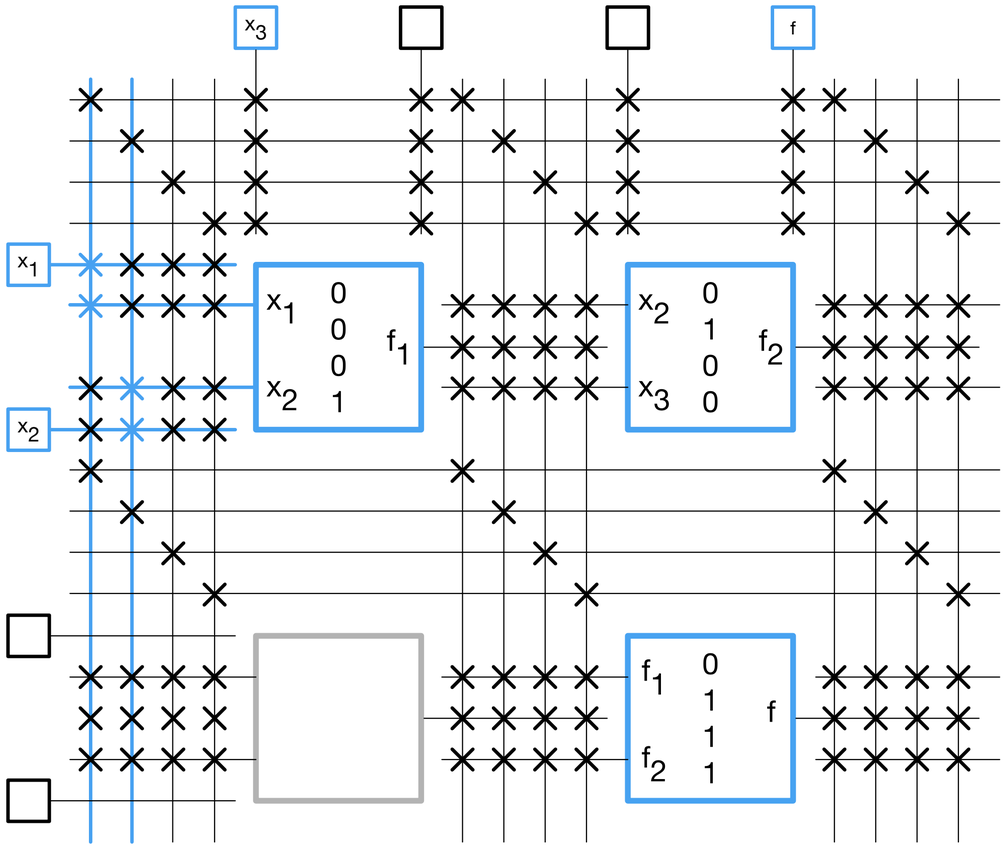
FPGA Compilation: Example

Step 3: Routing (connect x_1 to f_1)



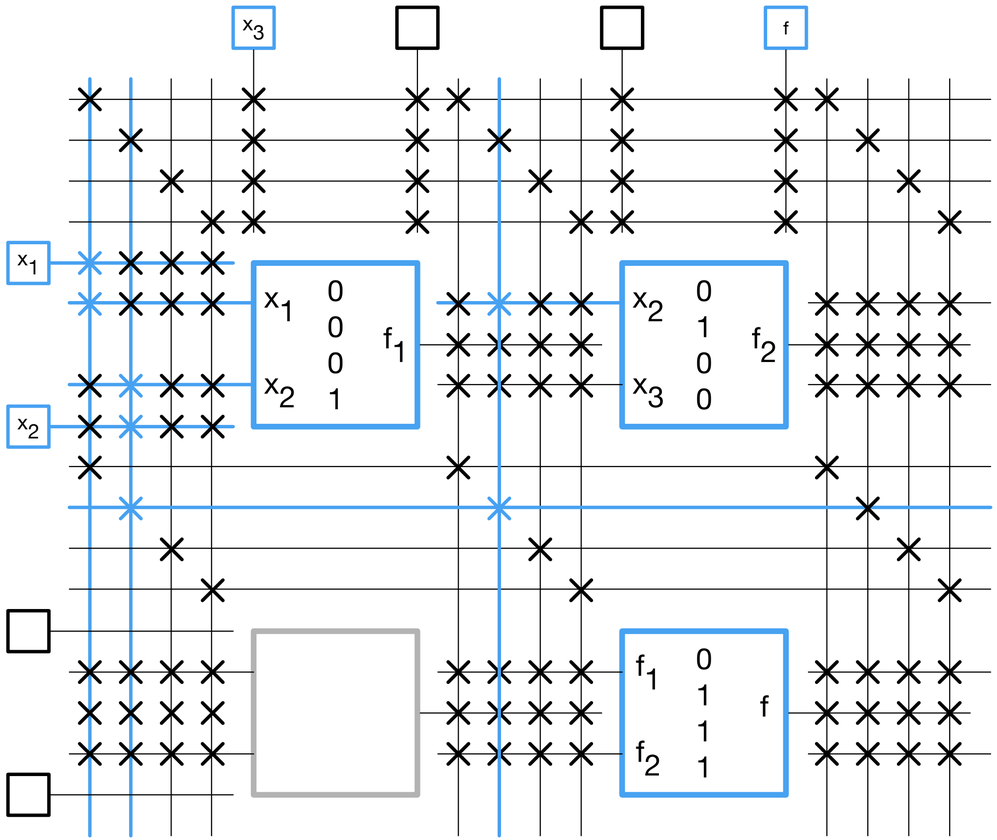
FPGA Compilation: Example

Step 3: Routing (connect x_2 to f_1)



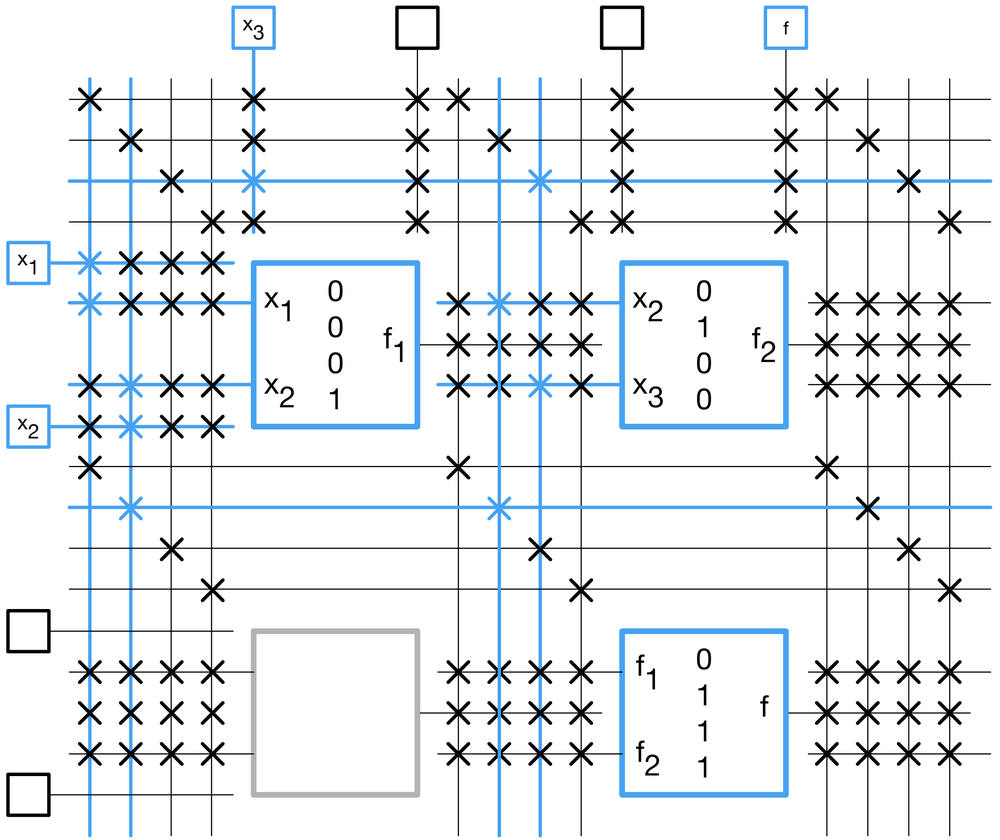
FPGA Compilation: Example

Step 3: Routing (connect x_2 to f_2)



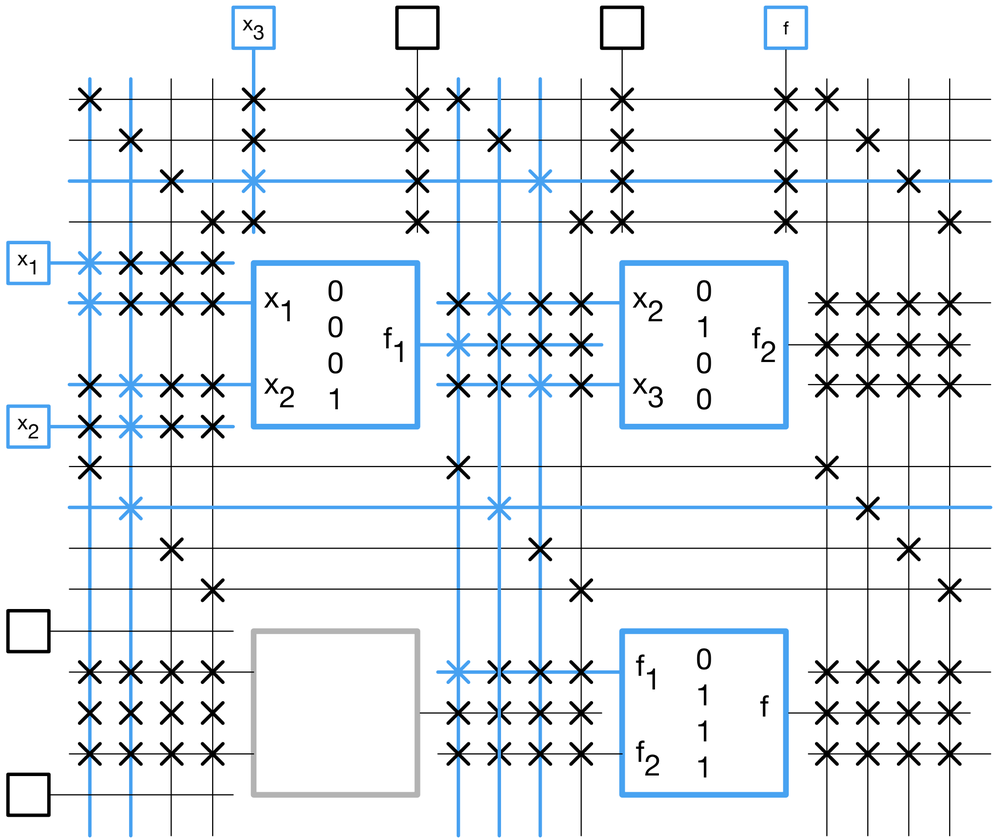
FPGA Compilation: Example

Step 3: Routing (connect x_3 to f_2)



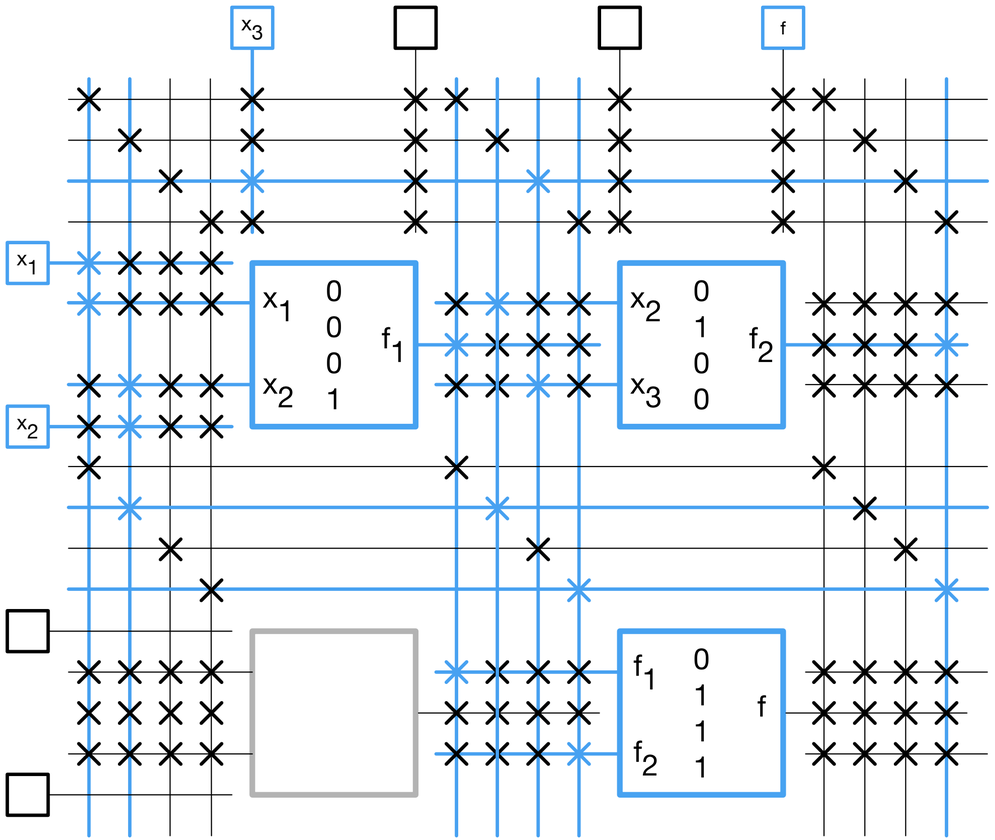
FPGA Compilation: Example

Step 3: Routing (connect f_1 to f)



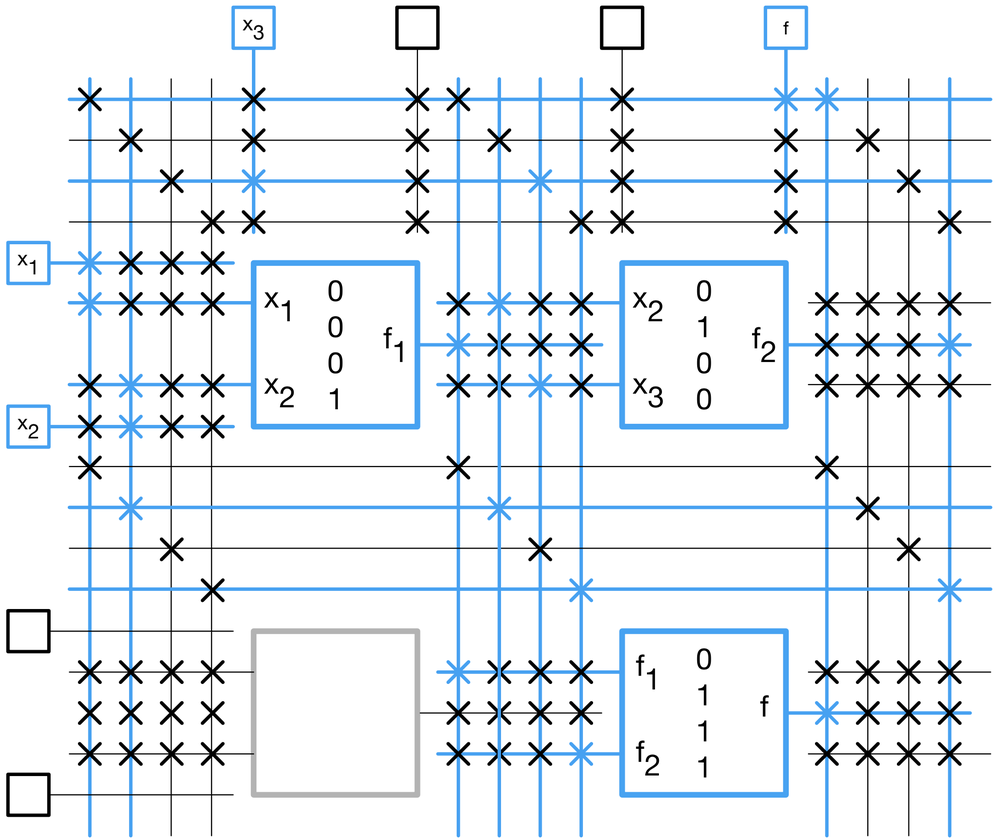
FPGA Compilation: Example

Step 3: Routing (connect f_2 to f)



FPGA Compilation: Example

Step 3: Routing (connect f to pin)

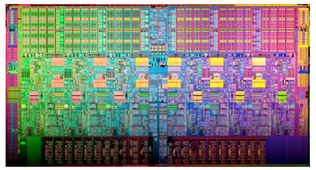


Talk Overview

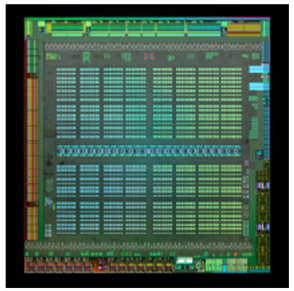
- What are FPGAs?
- How do you program FPGAs?
- **What can you do with FPGAs?**

FPGAs provide a high degree of specialization

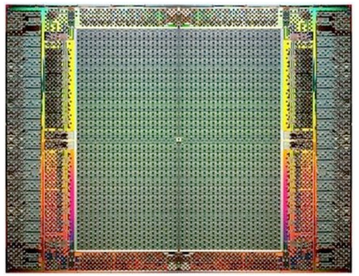
CPU



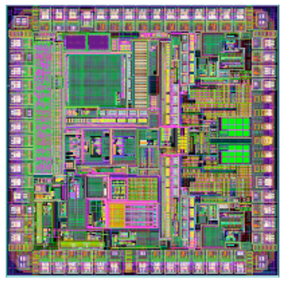
GPU



FPGA



ASIC



higher specialization & power efficiency **

Specialization for Deep Learning

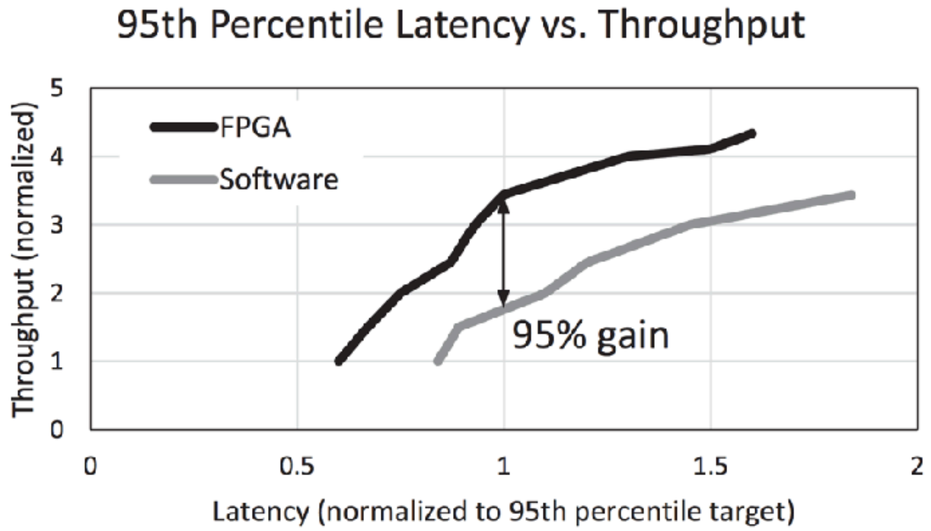
Zynq UltraScale+ can achieve 66 TOPS of binary operations,
a 2.5x increase over the Nvidia K-40 GPU [1]

Stratix 10 FPGA provide 2.3x better performance per watt over
a Titan X Pascal GPU with Ternary ResNet Deep Neural Net [2]

[1] Umuroglu et al., FINN: A Framework for Fast, Scalable, Binary Network Inference. [FPGA17]
[2] Nurvitadhi et al., Can FPGAs Beat GPUs in Accelerating Next-Generation Deep Neural Networks? [FPGA17]

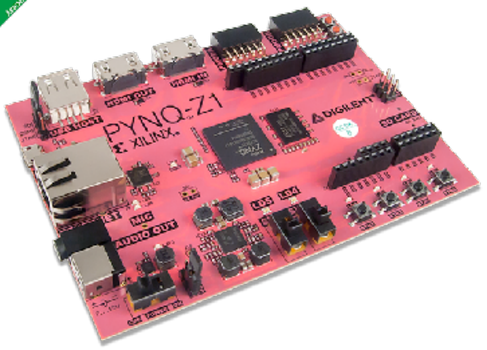
Specialization in the Datacenter

FPGAs have been used in deployment to accelerate Bing search among other things at Microsoft *



come to Andrew's talk on how they achieved this on Friday!

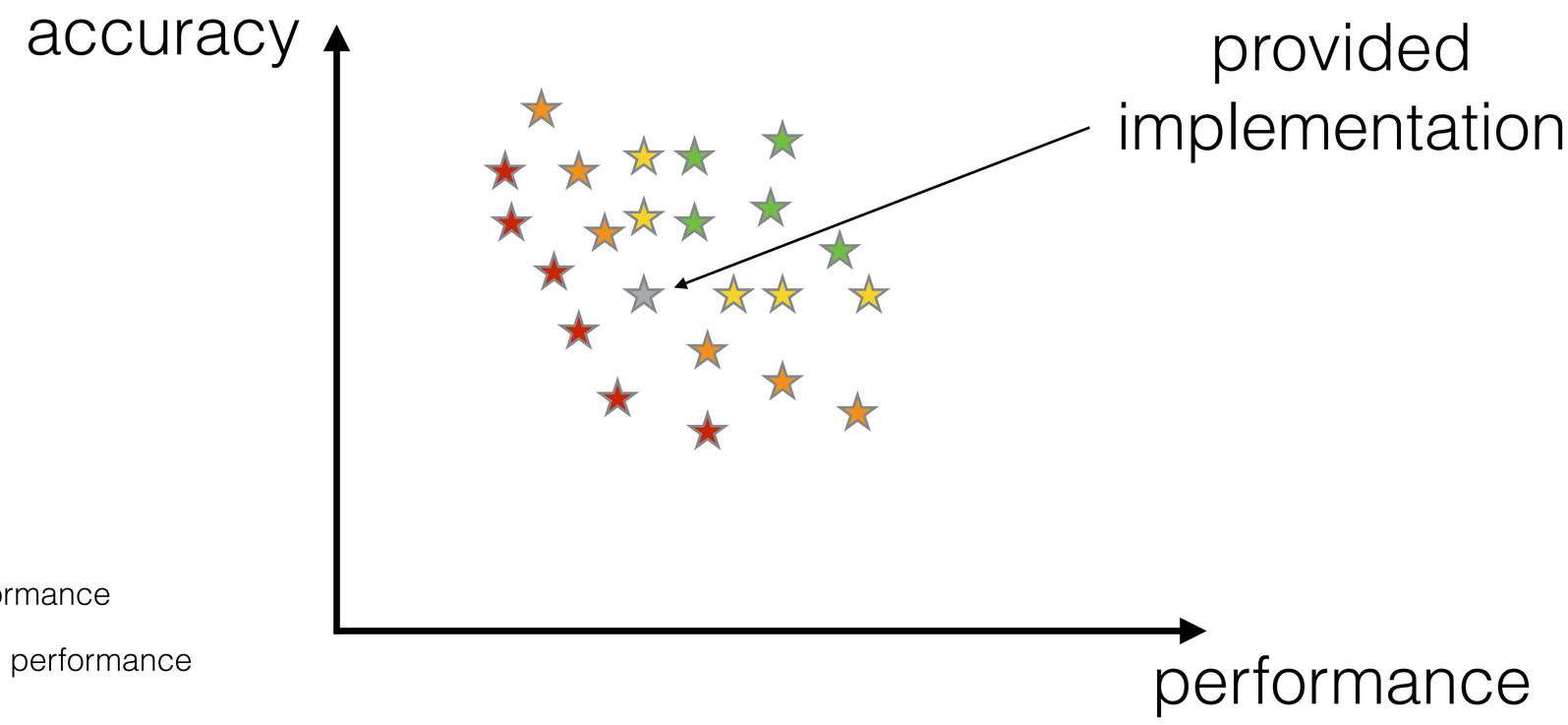
* Putnam et al., A Reconfigurable Fabric for Accelerating Large-Scale Datacenter Services? [ISCA14]



Homework 3 Overview

- You will implement your own ML classifier on a Zynq FPGA!
- Each one of you will get a PYNQ board and peripherals.
- I will provide you with scaffolding to generate a hardware design (overlay) with HLS, and driver libraries to test your design on the PYNQ.
- You will get to implement your own optimization that either improves performance, or accuracy from the baseline design.

Homework 3: Pareto-Optimality



Legend

- worse accuracy and performance
- better accuracy but worse performance
  or
  better performance but worse accuracy
- same accuracy but better performance or same performance but better accuracy
- better accuracy and better performance

Questions