

Hardware Modeling [VU] (191.011)

– WS25 –

Logic Synthesis, Place and Route

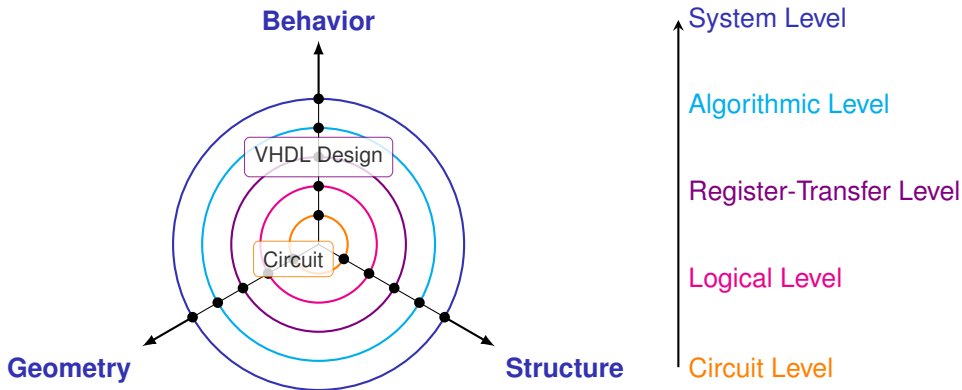
Florian Huemer & Sebastian Wiedemann & Dylan Baumann

WS 2025/26

Recall: Y-Diagram

HWMod
WS25

- Tools translate between different views



Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

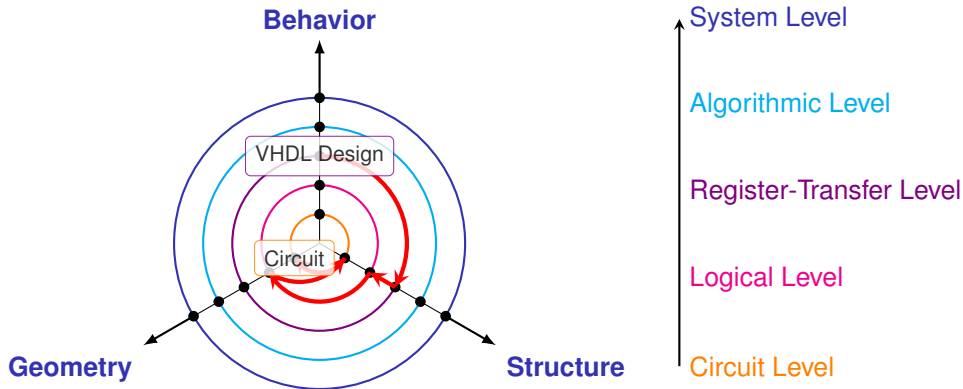
FPGAs

Place & Route

Recall: Y-Diagram

HWMod
WS25

- Tools translate between different views
 - VHDL design to circuit implementation



Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

FPGAs

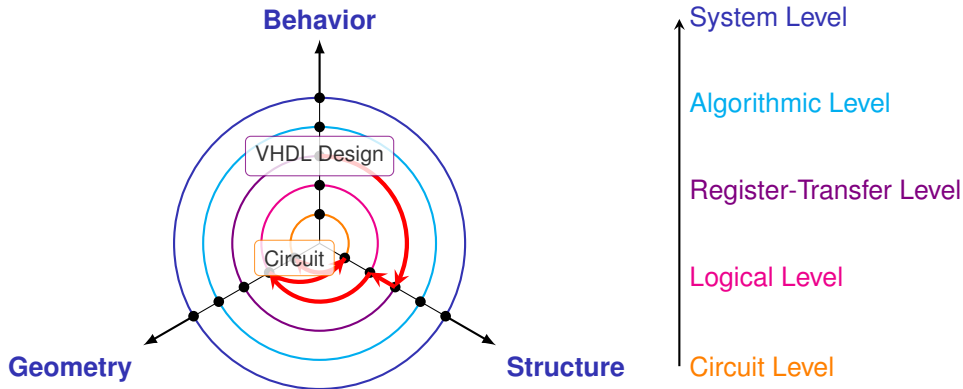
Place & Route

Recall: Y-Diagram

HWMod
WS25

Syn. & PR
Introduction
Design Entry
Compilation
Tech. Map.
FPGAs
Place & Route

- Tools translate between different views
 - VHDL design to circuit implementation
- What do the tools do? How do translations work?



Hardware Design Flow

HWMod
WS25

Syn. & PR

Introduction

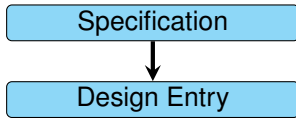
Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route



verbal / formal description of behavior

model in description language (e.g., VHDL)

Hardware Design Flow

HWMod
WS25

Syn. & PR

Introduction

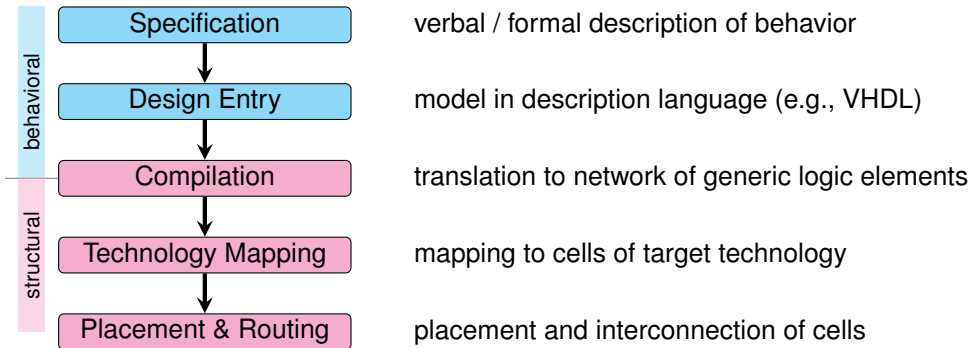
Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route



Hardware Design Flow

HWMod
WS25

Syn. & PR

Introduction

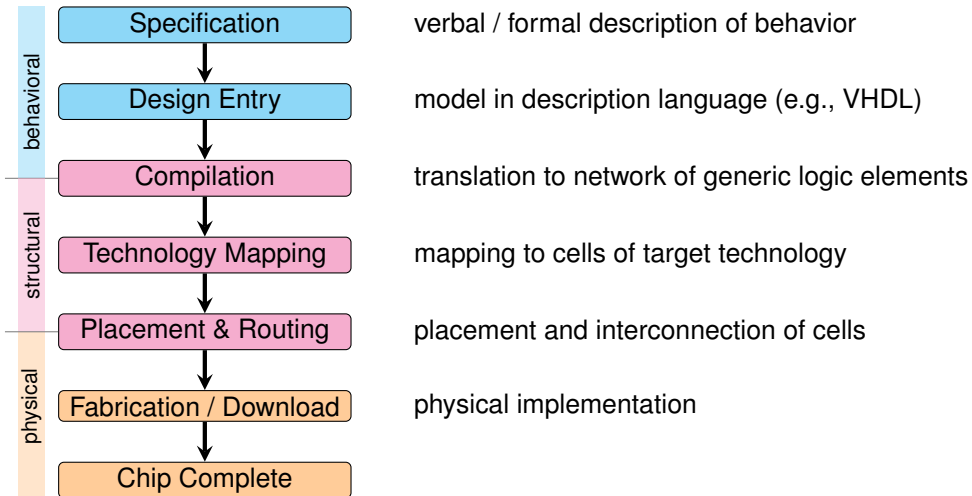
Design Entry

Compilation

Tech. Map.

FPGAs

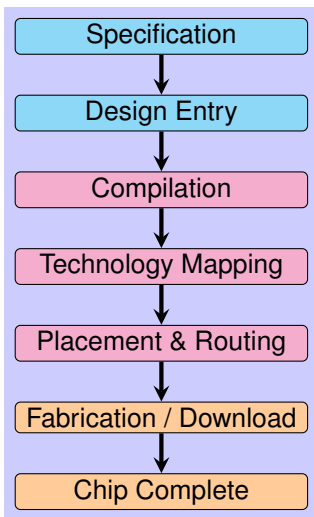
Place & Route



Hardware Design Flow (cont'd)

HWMod
WS25

Implementation



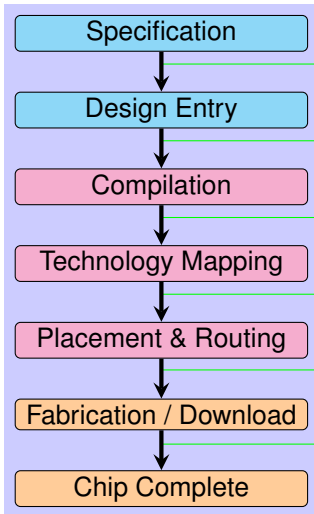
Syn. & PR
Introduction
Design Entry
Compilation
Tech. Map.
FPGAs
Place & Route

Hardware Design Flow (cont'd)

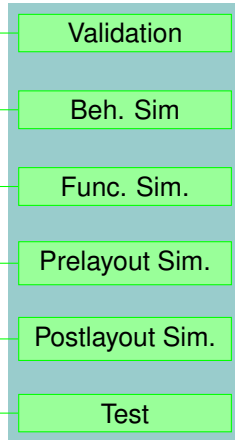
HWMod
WS25

Syn. & PR
Introduction
Design Entry
Compilation
Tech. Map.
FPGAs
Place & Route

Implementation



Verification



Hardware Design Flow (cont'd)

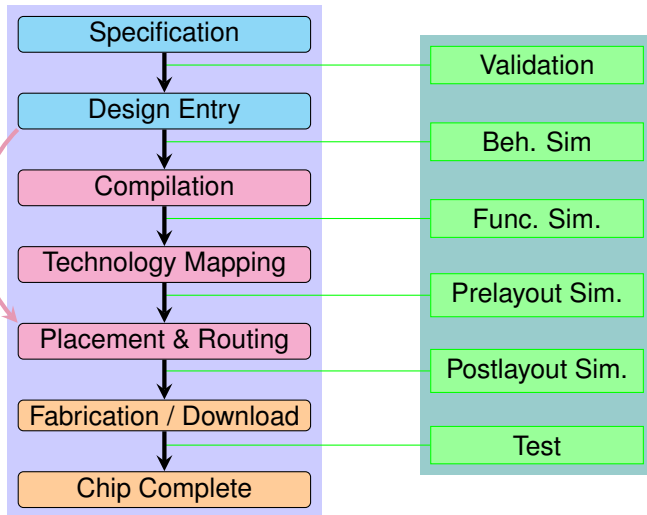
HWMoD
WS25

Syn. & PR
Introduction
Design Entry
Compilation
Tech. Map.
FPGAs
Place & Route

Implementation

Synthesis

Verification



Design Entry

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route

- Description of target circuit (behavioral RTL in VHDL)

Design Entry

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route

- Description of target circuit (behavioral RTL in VHDL)
 - Readable by tools, basis for simulation and documentation

Design Entry

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map

FPGAs

Place & Route

- Description of target circuit (behavioral RTL in VHDL)
 - Readable by tools, basis for simulation and documentation
- Example: Synchronous counter

```
1 entity counter is
2   port (
3       clk, res_n : in std_ulogic;
4       number      : out unsigned(7 downto 0)
5   );
6 end entity;
7
8 architecture beh of counter is begin
9   sync : process (clk, res_n) begin
10      if res_n = '0' then
11          number <= (others => '0');
12      elsif rising_edge(clk) then
13          number <= number + 1;
14      end if;
15   end process;
16 end architecture;
```

Design Entry

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map

FPGAs

Place & Route

- Description of target circuit (behavioral RTL in VHDL)
 - Readable by tools, basis for simulation and documentation
- Example: Synchronous counter

```
1 entity counter is
2   port (
3       clk, res_n : in std_ulogic;
4       number      : out unsigned(7 downto 0)
5   );
6 end entity;
7
8 architecture beh of counter is begin
9   sync : process (clk, res_n) begin
10      if res_n = '0' then
11          number <= (others => '0');
12      elsif rising_edge(clk) then
13          number <= number + 1;
14      end if;
15   end process;
16 end architecture;
```

Design Entry

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map

FPGAs

Place & Route

- Description of target circuit (behavioral RTL in VHDL)
 - Readable by tools, basis for simulation and documentation
- Example: Synchronous counter

```
1 entity counter is
2   port (
3       clk, res_n : in std_ulogic;
4       number      : out unsigned(7 downto 0)
5   );
6 end entity;
7
8 architecture beh of counter is begin
9   sync : process (clk, res_n) begin
10      if res_n = '0' then
11          number <= (others => '0');
12      elsif rising_edge(clk) then
13          number <= number + 1;
14      end if;
15  end process;
16 end architecture;
```

Compilation

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route

- Design entry converted to **technology-agnostic** *netlist*

Compilation

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route

- Design entry converted to **technology-agnostic** *netlist*
- Netlist: circuit as graph \Leftrightarrow RTL schematic

Compilation

HWMod
WS25

Syn. & PR

Introduction

Design Entry

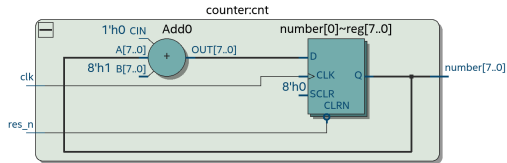
Compilation

Tech. Map.

FPGAs

Place & Route

- Design entry converted to **technology-agnostic netlist**
- Netlist: circuit as graph \Leftrightarrow RTL schematic



Compilation

HWMod
WS25

Syn. & PR

Introduction

Design Entry

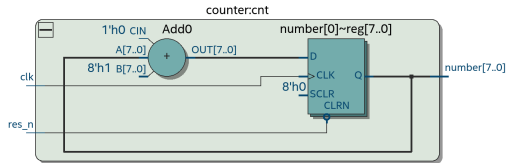
Compilation

Tech. Map.

FPGAs

Place & Route

- Design entry converted to **technology-agnostic netlist**
- Netlist: circuit as graph \Leftrightarrow RTL schematic
- Generic components (MUX, adder, basic gates)



Compilation

HWMod
WS25

Syn. & PR

Introduction

Design Entry

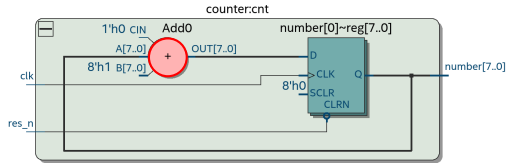
Compilation

Tech. Map.

FPGAs

Place & Route

- Design entry converted to **technology-agnostic netlist**
- Netlist: circuit as graph \Leftrightarrow RTL schematic
- Generic components (MUX, adder, basic gates)
 - Tools determine interfaces and check connections



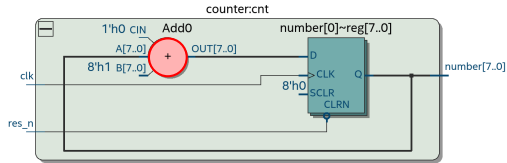
Compilation

HWMod
WS25

Syn. & PR

Introduction
Design Entry
Compilation
Tech. Map.
FPGAs
Place & Route

- Design entry converted to **technology-agnostic netlist**
- Netlist: circuit as graph \Leftrightarrow RTL schematic
- Generic components (MUX, adder, basic gates)
 - Tools determine interfaces and check connections
 - Might not exist in target technology



Technology Mapping

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route

- Map generic elements to target technology

Technology Mapping

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route

- Map generic elements to target technology
 - E.g.: ASIC, PAL, Gate-Array, **FPGA**

Technology Mapping

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route

- Map generic elements to target technology
 - E.g.: ASIC, PAL, Gate-Array, **FPGA**
- Requires *target library* of available cells
 - Gates, sequential elements, etc.
- Target technology is now fixed

Introduction to FPGAs

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route

■ **Field Programmable Gate Array**

Introduction to FPGAs

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route

■ Field Programmable Gate Array ?!

Introduction to FPGAs

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route

- **Field Programmable Gate Array** ?!
- Generic circuit, configurable for target circuit

Introduction to FPGAs

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route

- **Field Programmable Gate Array** ?!
- Generic circuit, configurable for target circuit
- Programmable logic cells and interconnect

Introduction to FPGAs

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route

- **Field Programmable Gate Array ?!**
- Generic circuit, configurable for target circuit
- Programmable logic cells and interconnect
 - + Large volumes \Rightarrow comparably cheap

Introduction to FPGAs

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route

- **Field Programmable Gate Array ?!**
- Generic circuit, configurable for target circuit
- Programmable logic cells and interconnect
 - + Large volumes \Rightarrow comparably cheap
 - + Already fabricated \Rightarrow simpler, faster design

Introduction to FPGAs

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route

- **Field Programmable Gate Array ?!**
- Generic circuit, configurable for target circuit
- Programmable logic cells and interconnect
 - + Large volumes \Rightarrow comparably cheap
 - + Already fabricated \Rightarrow simpler, faster design
 - Already fabricated \Rightarrow less optimizations

Introduction to FPGAs

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route

- **Field Programmable Gate Array ?!**
- Generic circuit, configurable for target circuit
- Programmable logic cells and interconnect
 - + Large volumes \Rightarrow comparably cheap
 - + Already fabricated \Rightarrow simpler, faster design
 - Already fabricated \Rightarrow less optimizations
 - Generic design \Rightarrow inevitable overhead

Introduction to FPGAs

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route

- **Field Programmable Gate Array ?!**
- Generic circuit, configurable for target circuit
- Programmable logic cells and interconnect
 - + Large volumes \Rightarrow comparably cheap
 - + Already fabricated \Rightarrow simpler, faster design
 - Already fabricated \Rightarrow less optimizations
 - Generic design \Rightarrow inevitable overhead
- How can such circuits be built?

Look-Up Tables (LUTs)

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route

- How to implement arbitrary logic functions?
 - Map inputs to outputs \Rightarrow truth table

a	b	y
0	0	1
0	1	1
1	0	0
1	1	0

Look-Up Tables (LUTs)

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

FPGAs

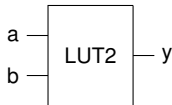
Place & Route

■ How to implement arbitrary logic functions?

■ Map inputs to outputs \Rightarrow truth table

\Rightarrow Same in FPGAs: *Look-up Tables* (LUTs)

a	b	y
0	0	1
0	1	1
1	0	0
1	1	0



Look-Up Tables (LUTs)

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route

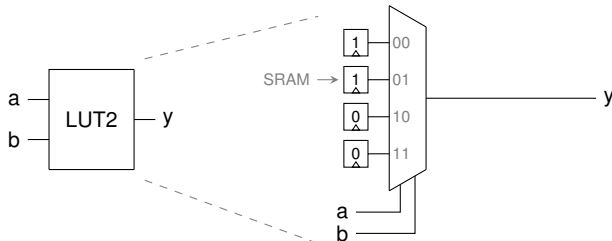
■ How to implement arbitrary logic functions?

- Map inputs to outputs \Rightarrow truth table

\Rightarrow Same in FPGAs: *Look-up Tables* (LUTs)

- Store in SRAM cells, use inputs to select

<i>a</i>	<i>b</i>	<i>y</i>
0	0	1
0	1	1
1	0	0
1	1	0



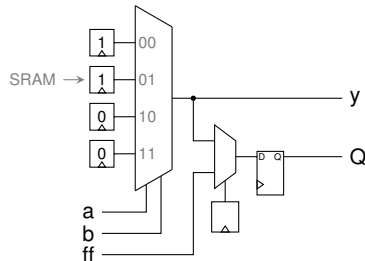
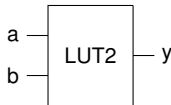
Look-Up Tables (LUTs)

HWMod
WS25

Syn. & PR
Introduction
Design Entry
Compilation
Tech. Map.
FPGAs
Place & Route

- How to implement arbitrary logic functions?
 - Map inputs to outputs \Rightarrow truth table
 - \Rightarrow Same in FPGAs: *Look-up Tables* (LUTs)
 - Store in SRAM cells, use inputs to select
- For sequential logic, add flip-flops

<i>a</i>	<i>b</i>	<i>y</i>
0	0	1
0	1	1
1	0	0
1	1	0



Look-Up Tables (LUTs)

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

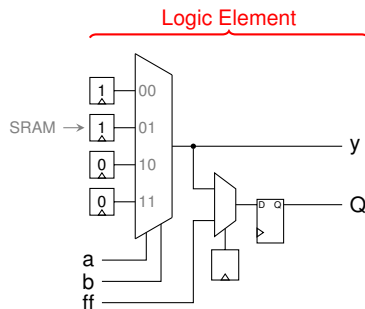
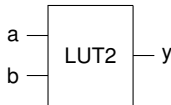
Tech. Map.

FPGAs

Place & Route

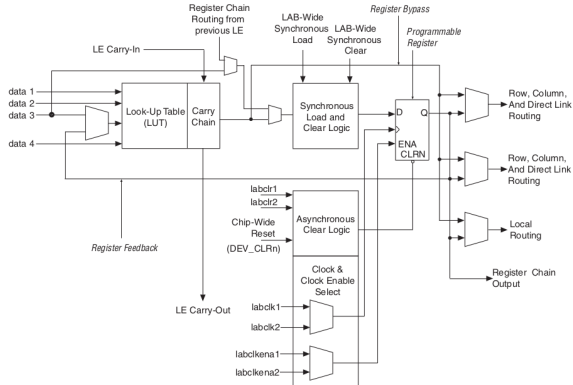
- How to implement arbitrary logic functions?
 - Map inputs to outputs \Rightarrow truth table
 - \Rightarrow Same in FPGAs: *Look-up Tables* (LUTs)
 - Store in SRAM cells, use inputs to select
- For sequential logic, add flip-flops
- *Logic Element* (LE)

<i>a</i>	<i>b</i>	<i>y</i>
0	0	1
0	1	1
1	0	0
1	1	0



LEs in Real FPGAs

- Real LEs are more complex and powerful



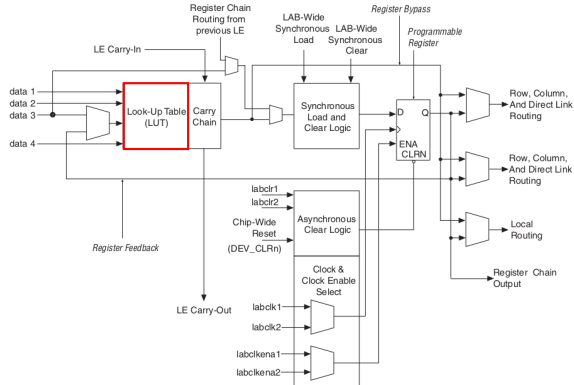
LEs in Real FPGAs

HwMod
WS25

- Real LEs are more complex and powerful
- Bigger LUTs (3-6 inputs)

Syn. & PR

Introduction
Design Entry
Compilation
Tech. Map.
FPGAs
Place & Route



LEs in Real FPGAs

HWMod
WS25

- Real LEs are more complex and powerful
- Bigger LUTs (3-6 inputs), dedicated carry chains

Syn. & PR

Introduction

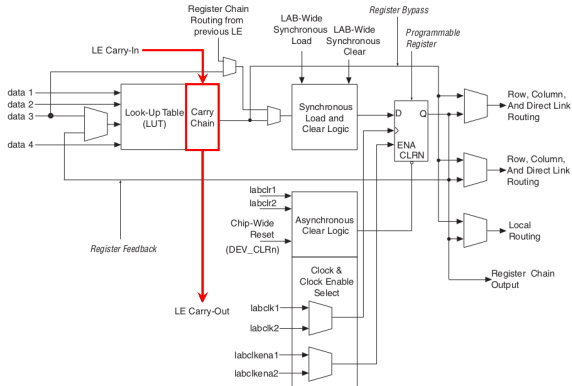
Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route



LEs in Real FPGAs

HWMod
WS25

Syn. & PR

Introduction

Design Entry

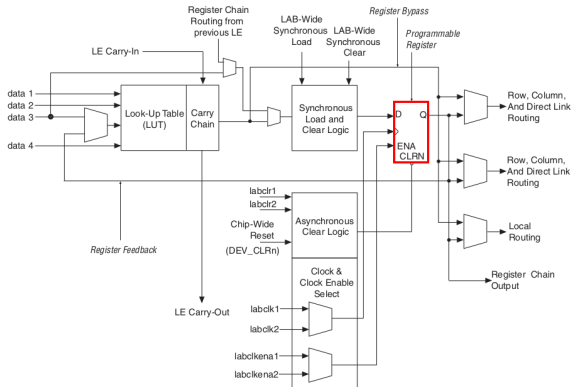
Compilation

Tech. Map.

FPGAs

Place & Route

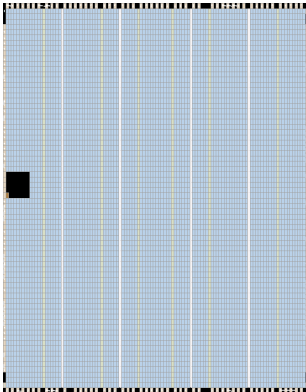
- Real LEs are more complex and powerful
- Bigger LUTs (3-6 inputs), dedicated carry chains
- Clear/set logic, sometimes multiple FFs



FPGA Structure

HWMod
WS25

- FPGA: grid of LE clusters and other elements



Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

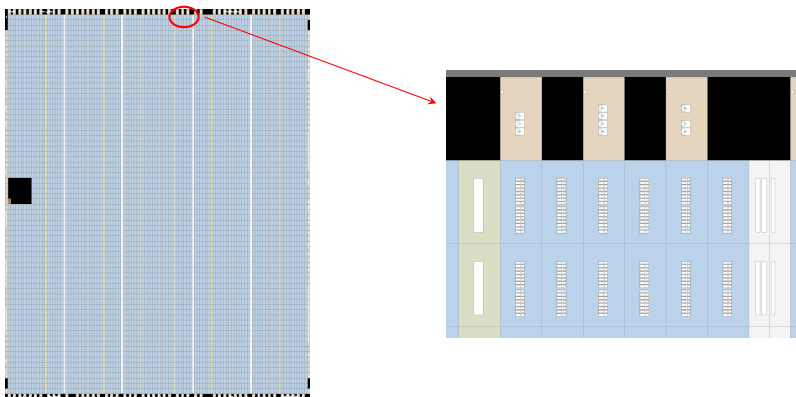
FPGAs

Place & Route

FPGA Structure

HWMod
WS25

- FPGA: grid of LE clusters and other elements



Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

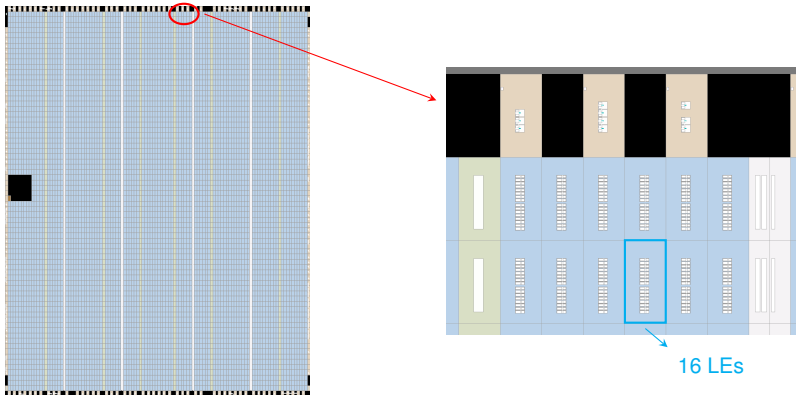
FPGAs

Place & Route

FPGA Structure

HWMod
WS25

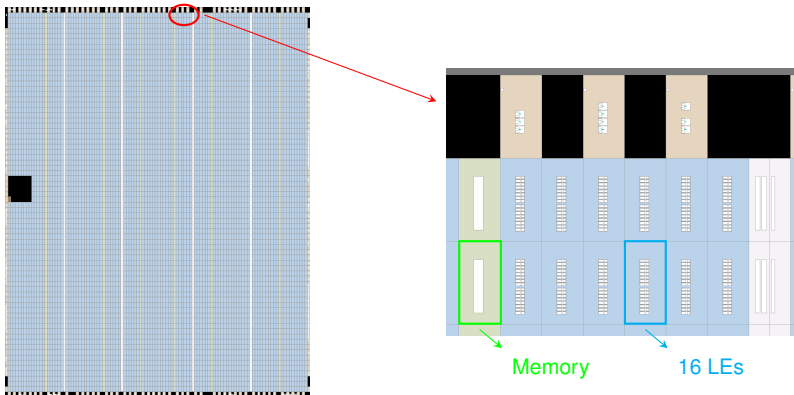
- FPGA: grid of LE clusters and other elements



FPGA Structure

HWMod
WS25

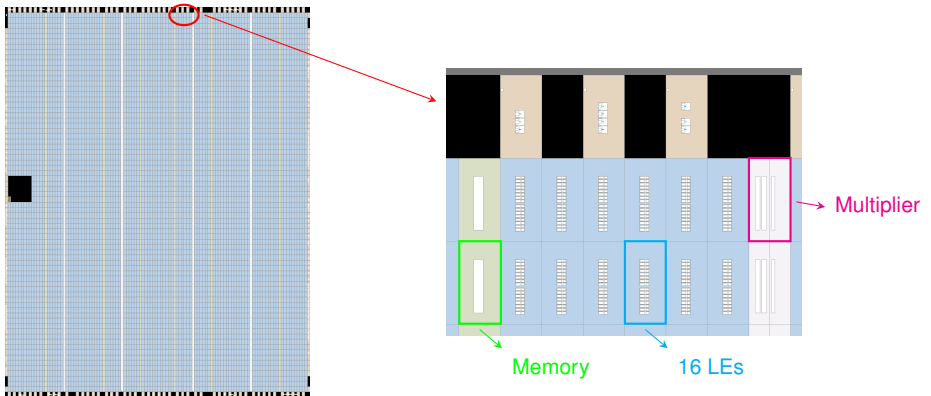
- FPGA: grid of LE clusters and other elements



FPGA Structure

HWMod
WS25

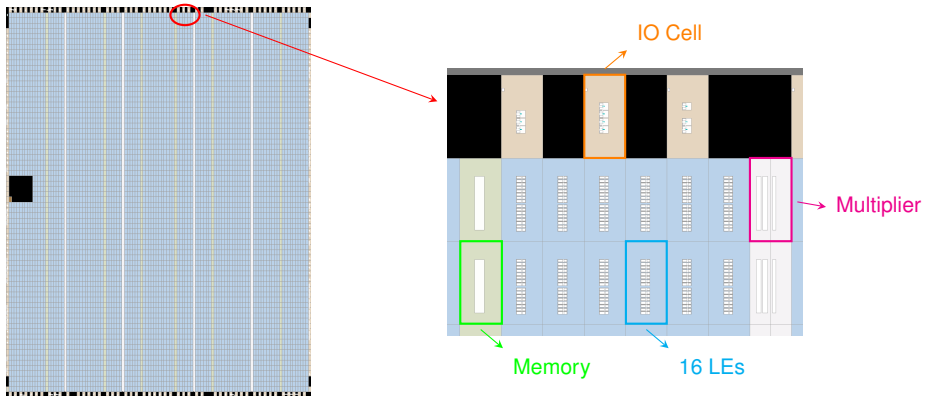
■ FPGA: grid of LE clusters and other elements



FPGA Structure

HWMod
WS25

■ FPGA: grid of LE clusters and other elements



FPGA Programmable Interconnect

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route

- How to connect LEs to form complex circuits?

FPGA Programmable Interconnect

HWMod
WS25

Syn. & PR

Introduction

Design Entry

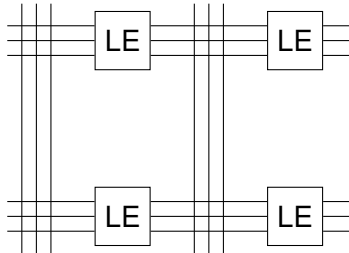
Compilation

Tech. Map.

FPGAs

Place & Route

- How to connect LEs to form complex circuits?
⇒ Programmable interconnect



FPGA Programmable Interconnect

HWMod
WS25

Syn. & PR

Introduction

Design Entry

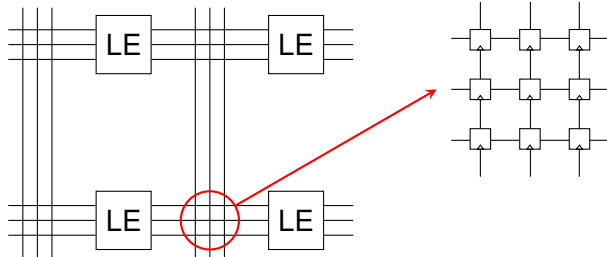
Compilation

Tech. Map.

FPGAs

Place & Route

- How to connect LEs to form complex circuits?
 - ⇒ Programmable interconnect
 - Programmable via SRAM (connected through MUXes)



Technology Mapping to an FPGA

HWMod
WS25

Syn. & PR

Introduction

Design Entry

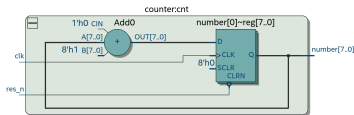
Compilation

Tech. Map.

FPGAs

Place & Route

- Generic elements implemented using LEs



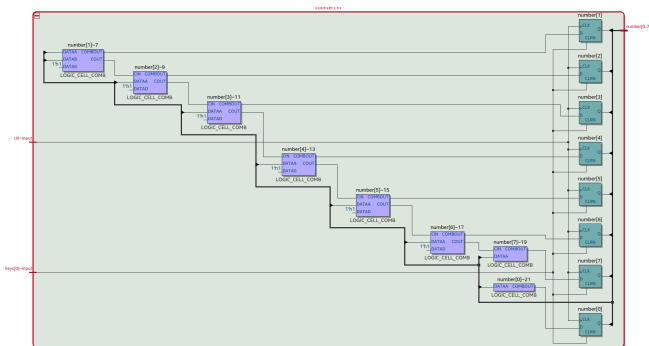
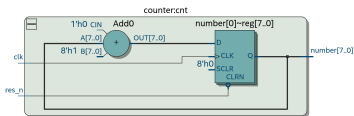
Technology Mapping to an FPGA

HWMod
WS25

- Generic elements implemented using LEs
- Each LE implements part of the logic

Syn. & PR

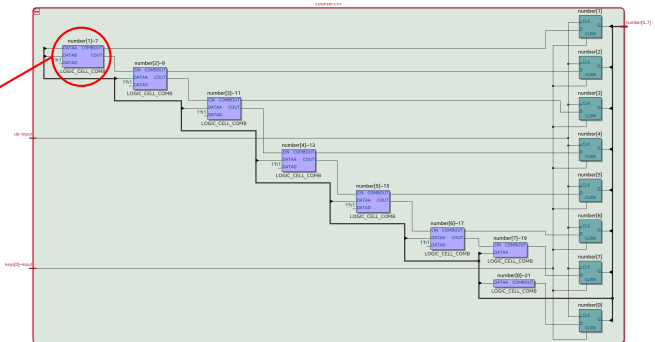
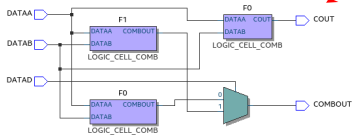
Introduction
Design Entry
Compilation
Tech. Map.
FPGAs
Place & Route



Technology Mapping to an FPGA

HWMod
WS25

- Generic elements implemented using LEs
- Each LE implements part of the logic
 - Here: carry chain for adders is used



Placement & Routing

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route

Placement & Routing

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route

- Placement: Choose position of LEs

Placement & Routing

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route

- Placement: Choose position of LEs
 - Constrained by physical availability

Placement & Routing

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route

- Placement: Choose position of LEs
 - Constrained by physical availability
 - Minimize unknown interconnect

Placement & Routing

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route

- Placement: Choose position of LEs
 - Constrained by physical availability
 - Minimize unknown interconnect
- Routing: Connect placed cells
 - Constrained by available interconnect
 - Goal: Meet timing constraints

Placement & Routing

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route

- Placement: Choose position of LEs
 - Constrained by physical availability
 - Minimize unknown interconnect
- Routing: Connect placed cells
 - Constrained by available interconnect
 - Goal: Meet timing constraints
- Paramount for timing, non-trivial
 - Multiple iterations
 - Driven by heuristics

Placement & Routing

HWMod
WS25

Syn. & PR

Introduction

Design Entry

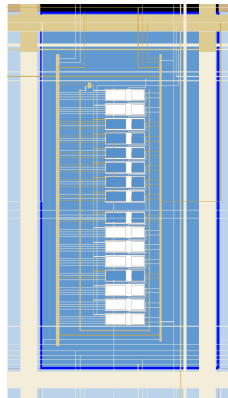
Compilation

Tech. Map.

FPGAs

Place & Route

- Placement: Choose position of LEs
 - Constrained by physical availability
 - Minimize unknown interconnect
- Routing: Connect placed cells
 - Constrained by available interconnect
 - Goal: Meet timing constraints
- Paramount for timing, non-trivial
 - Multiple iterations
 - Driven by heuristics



FPGA Bitstream

HWMod
WS25

Syn. & PR

Introduction

Design Entry

Compilation

Tech. Map.

FPGAs

Place & Route

- For FPGAs: result is a *bitstream*
 - Content of all SRAM cells
 - Configures LEs and interconnect

FPGA Bitstream

HwMod
WS25

Syn. & PR

Introduction

Design Entry

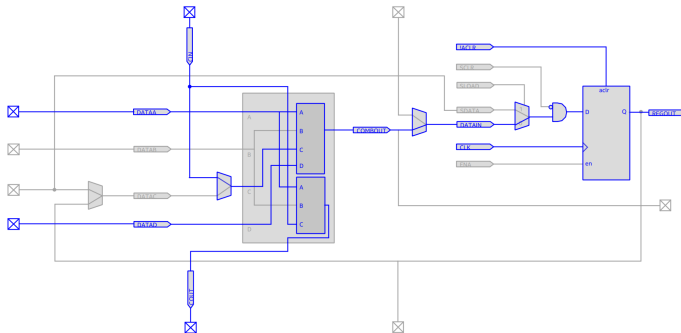
Compilation

Tech. Map.

FPGAs

Place & Route

- For FPGAs: result is a *bitstream*
 - Content of all SRAM cells
 - Configures LEs and interconnect



Lecture Complete!