CSCI 250 Introduction to Computer Organisation Lecture 3: CPU Architecture III



Jetic Gū 2025 Winter Semester (S1)



Overview

- Architecture: von Neumann
- Textbook: LCD: 9.7; CO: 2.1
- Core Ideas:
 - 1. Data bus ports in LogicWorks
 - 2. Lab 3 stuff

- Original Plan: more VHDL heavy implementations
- Reality:
 - The intended software (Vivado, ISE) are not properly installed on the college computer.
 - The old software LogicWorks, as I've found out, has very very limited VHDL support, insufficient for us to design a full CPU efficiently.
- Conclusion
 - We'll have to draw a lot of circuits. I of course will make sure there's at least one way of designing every component within reasonable time. This will mean that the final CPU design will be greatly simplified.

Changes in CSCI250



P1 Bus Ports

Data Bus Ports

Bus Ports Why no Bus Ports by Default?

- reasons.
- restricted when it comes to what can be connected.
- will need 2 bus ports: D16 and D3.

• There's no bus ports on any microchips, just individual 1 bit ports for obvious

• LogicWorks' built-in bus ports for VHDL is very difficult to use, you are very

• Fortunately, it can be resolved using custom bus ports. In CSCI250 Lab 3, you







Implementing a D16 Port



P1 Bus Ports

Implementing a D16 Port

 Primitive Type Create a subcircuit symbol, Create a subcircuit symbol a Import the port list from an o Set to Symbol Only type, needed.
Set to primitive type. Must Port Connector
Subcircuit Options Delete existing pins before Lock subcircuit by default s Browse
Messages

2. Go to Options - Subcircuit and Part Type, select Set to primitive type, under the scroll field, select Port Connector

Part Type

- but don't store a circuit with it yet. and select an open circuit to attach to it. open circuit, but don't attach the circuit. ever has a subcircuit. be used with caution!
- Ŧ
- adding pins defined in subcircuit to the subcircuit can't be opened
- subcircuit definition





Bus Ports Implementing a D16 Port



4. Go to Options - Add Pins, type D[0..15], press Add. You should the have a pin list like this.

Pin Number Pin Type Pin Function		
D D D D D D D D D D D D D D D D D D D	D12 D13 D14 D15	



▲ ×

Bus Ports Implementing a D16 Port - ×

Pin Number				
Ρ	Pin Type		Bus Internal	
Ρ	Pin Function		Bidirectional	•
	D			D
	DO			D
	D1			D
	D2			D
	D3			
	D4			
	D5			
	D6			
	D7			
	D8			
	D9			
	D10			
	D11			

5. Select pins D0-D15, change the pin function to Bidirectional





6. Draw a symbol like on the left, add a Bus pin. Double click the letter D, change the name to D16

Properties	
Visible Text Font	
NOTE: Attributes can only be set on a single pin at a time. Attributes	∑ D16
OK Cancel	



Bus Ports Implementing a D16 Port

1		Save Part As
1	Part name:	D16
	Destination Library:	
	7400.clf Connectors.CLF CSCI250.clf Discretes.CLF Pseudo Devices.CLF Simulation Gates.clf Simulation IO.clf Simulation Logic.clf Spice.CLF VHDLPrims.clf	
-	New Lib Ope	n Lib Save Cancel

7. Save the component as D16 in your library.



P1 Bus Ports



7. Now you can use it just like any other ports (you will need to name them), but it's just like a normal D0..15 bus.



P2 Lab 3 Stuff

Lab 3 Stuff

Register Array

P2 Lab 3 Stuff





Multiplexers









- ARM Architecture has different opcodes for Logical Unit operations and Arithmetic Unit operations
- The logical unit operations are called *Data-processing* instructions
- and compare instructions

- processing?lang=en
- 2. https://developer.arm.com/documentation/ddi0406/c/Application-Level-Architecture/Thumb-Instruction-Set-Encoding/16-bit-Thumb-instruction-encoding/Shift-immediate---add--subtract--move--and-compare?lang=en

• The arithmetic operations are called Shift (immediate), add, subtract, move,

1. https://developer.arm.com/documentation/ddi0406/c/Application-Level-Architecture/Thumb-Instruction-Set-Encoding/16-bit-Thumb-instruction-encoding/Data









- formatted quite differently
- processing?lang=en
- 2. https://developer.arm.com/documentation/ddi0406/c/Application-Level-Architecture/Thumb-Instruction-Set-Encoding/16-bit-Thumb-instruction-encoding/Shift-immediate---add--subtract--move--and-compare?lang=en

• Some CPU architecture uses just one Arithmetic Logical Unit design. For ARM, I find it easier to separate the AU and LU since their opcodes are

1. https://developer.arm.com/documentation/ddi0406/c/Application-Level-Architecture/Thumb-Instruction-Set-Encoding/16-bit-Thumb-instruction-encoding/Data







- OpcodeB is taking different portions of the instruction for the AU and LU For the **AU**, it is the 13-11 digits of the instruction. For the **LU**, it is the 9-6 digits of the instruction. We'll discuss instruction interpretation implementation in Lecture 4
- Mode is specific to AU's adder-subtractor makeup Opcode, but to avoid confusion here we name them separately.



It is the 10-9 digits of the instruction. In the ARM specification manual, it is combined with 13-11 to

• Immediate is a value that's embedded into the instruction, you need to pad zeroes in order to use it







- Because of the restrictions found in LogicWorks' VHDL implementation, it will be difficult and buggy to use AU and LU designed using VHDL
- For Lab 3, please use Circuit Diagrams
 - You will need to use hierarchical design and buses quite a lot to avoid massive circuit diagrams. If you do things correctly, it's not going to take much more time than VHDL.
 - Design a 16bit adder subtractor, start from there. Use everything you learned in CSCI150, this will feel good.

