CSCI E-93, Fall 2024: Computer Architecture

Prof. James L. Frankel
Harvard University

Version of 5:29 PM ET 17-Dec-2024

First Class Meeting on 9/3/2024

First Class Meeting Agenda

- Class website & Canvas, Zoom links
- Staff introductions
- Polls
- Student introductions
- Class website & class information
- Student actions: Order books, Say Hello!, Student Locations, HarvardKey, Cisco AnyConnect VPN, cscie93.dce.harvard.edu instance, order hardware
- Course problem set overview
- Problem Sets 0 & 1
- Digital electronics & Boolean logic

Class Website, Canvas, Zoom Links

- Our **class website** is located at URL: **https://cscie93.dce.harvard.edu/fall2024/**

- Please participate in the live stream and ask questions verbally using **Zoom** available in **Canvas** (**https://canvas.harvard.edu/courses/148636**) under the **Zoom** menu
- In addition, questions may be asked textually using **Zoom's Chat facility**

Zoom

- You are encouraged to turn on your video feed
  - This allows the course staff to better determine if students seem puzzled and/or if they have questions

- Class and section meetings are recorded
  - Students who are unable to attend a meeting for any reason are able to view recordings later
  - It's still better to participate in the live session so that questions can be asked and answered
  - Many students find that reviewing material later to fully appreciate the details presented during class – even if they participated in the live class – is very helpful

Staff Introductions

- Professor
  - James “Jamie” Frankel
- Teaching Assistants
  - Mark Ford
  - Stephen Benjamin

Quick Polls

- Goals and Interest for Class Enrollment (Multiple Choice)
- Class Participation
- Section Participation
- Class Expectations (Multiple Choice)

- You can choose to answer the polls anonymously

Student Introductions

- Please tell us a little about yourself
  - Where you're located
  - What you do when you're not at Harvard
  - Your technical background
  - Your out-of-work/school hobbies

Tour of Class Website

- At the top there are alerts in red
- Quick Links
- Links for streaming and videos
- Info about midterm exam, prerequisites, overview, bibliography, instructors and section, Ed Discussion wiki/forum, Say Hello!, your location, git & GitHub, grading, accessibility, plagiarizing, publishing/distributing course materials, outline/approximate schedule, hardware-related information, agenda for the upcoming class, slides used in class, questionnaire & problem sets, assorted links, link to the section home page

Meeting Times

- Section meets on Tuesdays in Room L01, 53 Church Street, Harvard Square, Cambridge, Massachusetts from 6:45 PM to 7:45 PM Eastern Time (ET) and in Zoom using the **Section: HELIX Classroom room**
  - This is immediately before class meets

- Class meets on Tuesdays in Room L01, 53 Church Street, Harvard Square, Cambridge, Massachusetts from 8:00 PM to 10:15 PM Eastern Time (ET) and in Zoom using the **Class: HELIX Classroom room**
  - Elongated class meeting time

- I will attempt to include a break during the class meetings *(but no guarantee because of scope of material to be presented)*

Section

- Required part of class

- Very important
  - Discusses concepts & issues that are not covered in class
  - Often gives a sketch of algorithms and approaches to be used in solving the problem sets
  - Adds enrichment on topics discussed in class/lecture

  - Great forum for a more interactive dialog

  - Is live streamed and also recorded

Class Website Review

- Questions?
  - Questions are always welcomed
    - Any questions now?
  - If there is limited time to answer questions, I'll let you know

- Review of Class Website
  - Midterm exam
  - Prerequisites
  - Overview
  - Required and optional books
  - The daily agenda (these slides) and all slides used in class
  - …

- Order books, if you have not already done so
  - Contemporary Logic Design, 2/e; Katz & Borriello
  - Computer Organization and Design: The Hardware/Software Interface, MIPS Edition, 6/e; Patterson & Hennessy
  - The Designer's Guide to VHDL, 3/e; Ashenden
- Somewhat limited online access is available to all of our books through our Library Reserves link in Canvas
  - Katz & Borriello and Patterson & Hennessy (and perhaps others) are available for just three hours at a time

Required Readings

- Refer to the Approximate Schedule section of the course website for required readings to be completed before each class meeting

Say Hello!, Student Locations, Harvard Key, Using cscie93.dce.harvard.edu

- Submit a video in Canvas under **Discussions** as a reply to my “Say Hello!” topic
- Please post your primary location using **Student Locations** facility in Canvas
- Ensure that your **Harvard Key** is established
- Ensure that you are able to VPN into Harvard using vpn.harvard.edu and Cisco AnyConnect
- Ensure that you have an account on our cscie93.dce.harvard.edu AWS instance
  - **Once your VPN connection is established**, login to cscie93.dce.harvard.edu using SSH/SFTP (SecureCRT & SecureFX) with your HarvardKey NetID as your login name and your HarvardKey password as your password
  - If you are unable to login to cscie93.dce.harvard.edu, you may need to synchronize your HarvardKey password by using a browser to visit https://key.harvard.edu/manage-account and then clicking on “Synchronize Password >” and following the instructions on the next screen

g.harvard.edu e-mail Address

- If you're interested, you can get a g.harvard.edu account that will give you a *logname*@g.harvard.edu e-mail address and access to Google Apps for Harvard
  - Get started at http://g.harvard.edu/
  - Note: Please be aware that when you claim your g.harvard account that g.harvard will become your primary Harvard e-mail address. All official communication from Harvard will be sent to your new g.harvard address and your g.harvard account will become your HarvardKey login name.

Class Discussion Group: Ed Discussion

- Ask all non-personal questions in Ed so the whole class can benefit from the answers
  - Ed can be found in Canvas by following the Ed Discussion link (https://canvas.harvard.edu/courses/148636/external_tools/106629?display=borderless)
  - Students are welcome to answer questions there, too
  - Personal questions should be sent to the course staff via e-mail
    - If appropriate, include all three course staff members in e-mail to allow the fastest reply
  - All registered students should already be in our Ed group

- To **receive immediate notifications about new threads:** in our Ed group, in the upper right, click on the **Account** icon, then…
  - Select **Settings**
  - Click on **Notifications**
  - Change the frequency to receive e-mails about new threads to **Instant**
  - Also, ensure that all other **Notification Emails** are active

Intel/Altera/Terasic DE2-115 Hardware

- Show class the hardware
  - All students should order the hardware **immediately**
    - **Intel/Altera DE2-115 FPGA kit**
    - Note: Terasic has announced the discontinuation of the DE2-115 board. Because of the number of slide switches, pushbuttons, LEDs, seven-segment LEDs, etc., this board is crucial for this course. Therefore, order the hardware immediately.
    - Available from **Terasic** (http://www.terasic.com.tw/en/ & https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=165&No=502)
    - Academic pricing is available (US$779 usual; US$423 academic) – Great deal!
    - Valid student ID card or a screen shot that shows that you're registered in the course is sufficient to get the academic pricing
  - Students who are unable to order the hardware, may borrow hardware for the duration of the semester, but **must return the hardware promptly** after the semester ends
  - In addition to Terasic FPGA board, order **USB to serial adapter** & **serial cable**
  - Optional **static dissipative devices**: mat, strap, ground point

Midterm Exam

- Our **midterm exam** will be available starting at **8:00 PM ET on October 15, 2024**
  - The exam must be started within 24 hours of the date & time above
  - The exam is three hours in length
  - The exam will be administered under Proctorio
- There will be **no class meeting on October 15th**, but **section will still be held on October 15th**
  - No topics relevant to the midterm exam will be discussed in that section meeting

Problem Set Overview

- Problem Set 0: the course questionnaire, fix-this-program & word-count
- Problem Set 1: textbook problems from Katz & Borriello
- Problem Set 2: block diagram & instruction set
- Problem Set 3: VHDL counter, textbook problems from Katz & Borriello, final program in the C Programming Language
- Problem Set 4: assembler, final program in your assembly language
- Preliminary Final Project Problem Set (ALU)
- Problem Set 5: emulator
- Problem Set 6: sequencer action description, VHDL memory system interaction
- Final Project: VHDL processor design

Five Free Late Days

- Please don’t use any of your five free late days early in the class
- Because the later problem sets are built upon earlier problem sets, the free late days are more valuable later in the semester
- Also, the larger problem sets are worth more points and take much more time to complete

Problem Set 0

- Complete Problem Set 0
  - Establish a GitHub account
  - Install git as described on the section website (https://cscie93.dce.harvard.edu/fall2024/section/index.html)
  - Modify the course questionnaire with your personal answers
  - Fix warnings and errors in fix-this-program on the cscie93 instance
  - Write the word count program
  - Create a branch named “problem-set-0”, create a merge request, add the appropriate comment
- Due this coming Sunday night, September 8th, 2024 at midnight ET

Problem Set 1

- Present **Problem Set 1**
  - Due at midnight ET on Sunday night, September 15th, 2024

Lying to Students

- I will lie to you this semester

Lying to Students

- I will lie to you this semester
  - There are too many details to give the whole truth
  - That is the only way we can make reasonable progress through the material

- By the end of the semester, all lies will be fully corrected

Non-academic Class Activities

- Encourage a student community

- If interested, students are welcome to gather with us after each class for dinner in Cambridge
  - Opportunity for students to socialize in an informal setting outside of class
  - Discussion/conversation/sharing after class
  - Not relevant to class

- Other non-class activities
  - Class ski trip in January

Class Break

- Let’s take a 10 minute break
- You’re welcome informally interact during the break

Today’s New Material

- Cover **Binary Logic Levels** slides
- Cover **Boolean Logic** slides through **NAND Gate** slide **#21**

Second Class Meeting on 9/10/2024

Second Class Meeting Agenda

- Questions and Comments
- Administrivia
  - Start work on problem sets early
  - Section
  - Status of Terasic orders
  - Intel/Altera/Terasic DE2-115 Hardware
  - Intel/Altera Quartus FPGA Design Software
  - Student Locations
  - Say Hello! in Canvas
  - Class Discussion Group: Ed Discussion
- Current Problem Set Status
- Boolean Logic (continued)
- Boolean Logic Continued
- Advanced Boolean Logic
- Laws and Theorems of Boolean Logic
- Gray Codes & Karnaugh Maps
- Computer Logic

Questions and Comments

- Section
- Last week's class
- Problem Sets 0 & 1
- Access to the class cscie93.dce.harvard.edu instance
- Ed Discussion threads
- Readings
- Anything else

Start Work on Problem Sets Early

- In order to ensure that your questions are answered in a timely manner, start work on problem sets early
- The course staff are often quite prompt in answering Ed questions, but there is no guarantee of immediate responses

Section

- Required part of class
- Section meets immediately before class on Tuesdays from 6:45 PM to 7:45 PM Eastern Time (ET) and in Zoom using the **Section: HELIX Classroom room**

- Very important
  - Discusses concepts & issues that are not covered in class
  - Often gives a sketch of algorithms and approaches to be used in solving the problem sets
  - Adds enrichment on topics discussed in class/lecture

  - Great forum for a more interactive dialog

  - Is live streamed and also recorded

Status of Terasic Orders

- Has everyone who wants to order hardware already done so?
- Problem Set 3 (due on October 13, 2024) is the first problem set that requires the hardware

- If you are not able or do not want to purchase hardware, I will be able to **distribute hardware** during class on **Tuesday, September 24, 2024** to students who are local to Cambridge/Boston
- If you want to borrow hardware for your use during the semester, you must let me know as soon as possible

Intel/Altera/Terasic DE2-115 Hardware

- Show class the hardware
  - All students should order the hardware **immediately**
    - **Intel/Altera DE2-115 FPGA kit**
    - Note: Terasic has announced the discontinuation of the DE2-115 board. Because of the number of slide switches, pushbuttons, LEDs, seven-segment LEDs, etc., this board is crucial for this course. Therefore, order the hardware immediately.
    - Available from **Terasic** (http://www.terasic.com.tw/en/ & https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=165&No=502)
    - Academic pricing is available (US$779 usual; US$423 academic) – Great deal!
    - Valid student ID card or a screen shot that shows that you're registered in the course is sufficient to get the academic pricing
  - Students who are unable to order the hardware, may borrow hardware for the duration of the semester, but **must return the hardware promptly** after the semester ends
  - In addition to Terasic FPGA board, order **USB to serial adapter** & **serial cable**
  - Optional **static dissipative devices**: mat, strap, ground point

Intel/Altera Quartus FPGA Design Software

- The most recent version of Quartus that supports the Altera DE2-115 boards (with a Cyclone IV FPGA) is Quartus Prime Lite Edition, Release 20.1.1
  - **All of us will be using Quartus Prime Lite Edition, Release 20.1.1**
- For release 20.1.1, only the Lite Edition can be utilized for free – it doesn't require a License Key
- Only 64-bit x86 processors running Windows are supported
  - You **are** able to use a VM under MacOS
  - You **are not** able to run Quartus under a VM on any Apple ARM (Apple Silicon) processors
  - Windows 10 is available for free from Microsoft
- **You need to have a 64-bit x86 processor or VM running Windows**

Student Locations

- Please post your primary location using **Student Locations** facility in Canvas

Say Hello! in Canvas

- Submit a video in Canvas under **Discussions** as a reply to my “Say Hello!” topic

Class Discussion Group: Ed Discussion

- Ask all non-personal questions in Ed so the whole class can benefit from the answers
  - Ed can be found in Canvas by following the Ed Discussion link (https://canvas.harvard.edu/courses/148636/external_tools/106629?display=borderless)
  - Students are welcome to answer questions there, too
  - Personal questions should be sent to the course staff via e-mail
    - If appropriate, include all three course staff members in e-mail to allow the fastest reply
  - All registered students should already be in our Ed group

- To **receive immediate notifications about new threads:** in our Ed group, in the upper right, click on the **Account** icon, then…
  - Select **Settings**
  - Click on **Notifications**
  - Change the frequency to receive e-mails about new threads to **Instant**
  - Also, ensure that all other **Notification Emails** are active

Current Problem Set Status

- **Problem Set 0 was due** this past Sunday night

- Go over **Problem Set 1**
  - All book problems from Katz and Borriello
  - Due at midnight ET on this coming Sunday, September 15th, 2024

- I will present **Problem Set 2** in class next week

Today's New Material

- Finish covering **Boolean Logic** slides
  - Cover **Boolean Logic** slides beginning with **NAND Gate** slide **#21**

- Cover new slides
  - Cover **Boolean Logic Continued** slides
  - Cover **Advanced Boolean Logic** slides
  - Cover **Laws and Theorems of Boolean Logic** slides
  - Cover **Gray Codes & Karnaugh Maps** slides
  - Cover **Computer Logic** slides

Third Class Meeting on 9/17/2024

Third Class Meeting Agenda (1 of 2)

- Questions and Comments
- Administrivia
  - Use of Generative AI
  - Start work on problem sets early
  - Section
  - Status of Terasic orders
    - All necessary hardware should have been received by now
  - Intel/Altera/Terasic DE2-115 Hardware
  - Intel/Altera Quartus FPGA Design Software
  - Library Reserves
  - Textbooks at the Coop
  - Student Locations
  - Say Hello! in Canvas
  - Class Discussion Group: Ed Discussion
- Review of Current Problem Set Status

Third Class Meeting Agenda (2 of 2)

- Computer Logic
- Place Values
- Numeric Encodings (Two’s Complement only)
- [Canonical Form, Minterms & Maxterms]
- [Dealing with Time in Combinational Circuits]
- MIPS Instruction Set
- MIPS Datapath - Single Memory - No Pipelining
- MIPS Coding Snippets

Questions and Comments

- Section
- Last week's class
  - Gates
  - Boolean Logic
  - Flip-Flops, D Latch
  - Composite Devices: Register, Mux, Decoder, Half and Full Adder, ALU
  - Sum-of-Products
  - Laws and Theorems of Boolean Logic
  - Gray Codes & Karnaugh Maps
  - Computer Logic/Block Diagram
- Problem Sets 0 & 1
- Ed Discussion threads
- Readings
- Anything else

Use of Generative AI

- We specifically forbid the use of ChatGPT or any other generative artificial intelligence (AI) tools at all stages of the work process, including preliminary ones
- It is the responsibility of each student to check with the course staff for any other exceptions to this policy
- Violations of this policy will be considered academic misconduct

Start Work on Problem Sets Early

- In order to ensure that your questions are answered in a timely manner, start work on problem sets early
- The course staff are often quite prompt in answering Ed questions, but there is no guarantee of immediate responses

Section

- Required part of class
- Section meets immediately before class on Tuesdays from 6:45 PM to 7:45 PM Eastern Time (ET) and in Zoom using the **Section: HELIX Classroom room**

- Very important
  - Discusses concepts & issues that are not covered in class
  - Often gives a sketch of algorithms and approaches to be used in solving the problem sets
  - Adds enrichment on topics discussed in class/lecture

  - Great forum for a more interactive dialog

  - Is live streamed and also recorded

Status of Terasic Orders

- Has anyone not received hardware yet?
- Problem Set 3 (due on October 13, 2024) is the first problem set that requires the hardware

- If you are not able or do not want to purchase hardware, I will be able to **distribute hardware** during **next week's class** on **Tuesday, September 24, 2024** to students who are local to Cambridge/Boston
- Hardware may be borrowed for the duration of the semester, but **must be returned promptly** when the semester ends
- You must let me know **via e-mail** **today or tomorrow** if you want to borrow hardware
  - So far, only one student has sent e-mail about borrowing hardware

Intel/Altera/Terasic DE2-115 Hardware

- All students should have already received the hardware
  - **Intel/Altera DE2-115 FPGA kit**
  - Note: Terasic has announced the discontinuation of the DE2-115 board. Because of the number of slide switches, pushbuttons, LEDs, seven-segment LEDs, etc., this board is crucial for this course. Therefore, order the hardware immediately.
  - Available from **Terasic** (http://www.terasic.com.tw/en/ & https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=165&No=502)
  - Academic pricing is available (US$779 usual; US$423 academic) – Great deal!
  - Valid student ID card or a screen shot that shows that you're registered in the course is sufficient to get the academic pricing
- In addition to Terasic FPGA board, order **USB to serial adapter** & **serial cable**
- Optional **static dissipative devices**: mat, strap, ground point

Intel/Altera Quartus FPGA Design Software

- The most recent version of Quartus that supports the Altera DE2-115 boards (with a Cyclone IV FPGA) is Quartus Prime Lite Edition, Release 20.1.1
  - **All of us will be using Quartus Prime Lite Edition, Release 20.1.1**
- For release 20.1.1, only the Lite Edition can be utilized for free – it doesn't require a License Key
- Only 64-bit x86 processors running Windows are supported
  - You **are** able to use a VM under MacOS
  - You **are not** able to run Quartus under a VM on any Apple ARM (Apple Silicon) processors
  - Windows 10 is available for free from Microsoft
- **You need to have a 64-bit x86 processor or VM running Windows**

Library Reserves

- Under Library Reserves in Canvas, our textbooks are available with ONLINE ACCESS

Textbooks at the Coop

- All course books are now available from the Harvard Coop

Student Locations

- Please post your primary location using **Student Locations** facility in Canvas

Say Hello! in Canvas

- Submit a video in Canvas under **Discussions** as a reply to my “Say Hello!” topic

Class Discussion Group: Ed Discussion

- Ask all non-personal questions in Ed so the whole class can benefit from the answers
  - Ed can be found in Canvas by following the Ed Discussion link (https://canvas.harvard.edu/courses/148636/external_tools/106629?display=borderless)
  - Students are welcome to answer questions there, too
  - Personal questions should be sent to the course staff via e-mail
    - If appropriate, include all three course staff members in e-mail to allow the fastest reply
  - All registered students should already be in our Ed group

- To **receive immediate notifications about new threads:** in our Ed group, in the upper right, click on the **Account** icon, then…
  - Select **Settings**
  - Click on **Notifications**
  - Change the frequency to receive e-mails about new threads to **Instant**
  - Also, ensure that all other **Notification Emails** are active

Problem Sets

- **Problem Set 1 was due** this past Sunday night

- Present **Problem Set 2**
  - Due at midnight ET on Sunday, September 29th, 2024

Comments on Problem Set 2 (1 of 2)

- Your instruction set description and block diagram will become the basis for the **Principles of Operation** documentation for your final project
- Your submission for PS2 will be continuously *revised throughout the semester*
- Certainly make your best effort to have a complete and correct submissions, but *don't worry if you still have open questions*
  - These will be resolved as the semester progresses

- Your best source for information needed to complete PS2 is from lecture, course slides, section, and readings
  - Please note the readings associated with class meetings in the Approximate Schedule section of the class website

Comments on Problem Set 2 (2 of 2)

- By default, design an instruction set that is MIPS-like, but with a 16-bit word size rather than a 32-bit word size
  - The smaller word size has implications across your entire design
    - How many registers will you support?
    - How much memory can you easily address?
    - How many different instruction opcodes can you support?
  - The change in word size makes the instruction set design an interesting project
- If you aren't implementing a MIPS-like machine, your design is probably more complex
  - Choosing an instruction set with which you're familiar will simplify the task
  - The MIPS instruction set is cleanly designed and relatively easy to understand

Today's New Material

- Review the **Computer Logic** slides
  - Show why all data paths exist in the block diagram

- Present new material
  - **Place Values** slides
  - **Numeric Encodings** slides (Two's Complement only)
  - [**Canonical Form, Minterms & Maxterms** slides]
  - [**Dealing with Time in Combinational Circuits** slides]
  - **MIPS Instruction Set** slides through the **JR Instruction Fields** slide **#51**

Fourth Class Meeting on 9/24/2024

Fourth Class Meeting Agenda

- Questions and Comments
- Administrivia
  - Intel/Terasic/Altera DE2-115 and Associated Hardware
- Review of Current Problem Set Status
- Intel/Altera Quartus Software
- USB Blaster Installation
- MIPS Instruction Set (continued)
- MIPS Datapath - Single Memory - No Pipelining
- MIPS Coding Snippets
- Numeric Encodings (Other than Two's Complement)
- Canonical Form, Minterms & Maxterms
- Dealing with Time in Combinational Circuits

Questions and Comments

- Section
- Last week's class
  - Gates
  - Boolean Logic
  - Flip-Flops, D Latch
  - Composite Devices: Register, Mux, Decoder, Half and Full Adder, ALU
  - Sum-of-Products
  - Laws and Theorems of Boolean Logic
  - Gray Codes & Karnaugh Maps
  - Computer Logic/Block Diagram
  - Two's Complement Integer Representation
  - MIPS Instruction Set
- Problem Sets 0, 1, and 2
- Ed Discussion threads
- Readings
- Anything else

Intel/Terasic/Altera DE2-115 and Associated Hardware

- Borrowed Intel/Terasic/Altera DE2-115 and Associated Hardware will be Distributed During Break

Problem Sets

- **Problem Set 2** is due at midnight ET this coming Sunday, September 29th, 2024
  - After PS2 is due and submitted, the course staff will meet with each student to review their computer instruction set and architecture through their Principles of Operation document

- Present **Problem Set 3**
  - Due at midnight ET on Sunday, October 13th, 2024

Problem Set 3 Overview

- VHDL 8-bit Counter
  - KEY2 counts up by one
  - KEY3 resets to zero
  - HEX7 & HEX6 seven-segment LEDs display the current counter value in hexadecimal
- Book Problems
- Final Program in the C Programming Language
  - It is required to use the specified function/procedure declarations
  - You're welcome to implement and include other functions/procedures

- A word to the wise: **Complete PS3 before the midterm exam**
  - **VHDL will not be covered** in the midterm exam

Intel/Altera Quartus Software

- We will be using Intel Quartus Prime *Lite Edition* Design Software, Version 20.1.1 for Windows
  - ID 660907
  - Release date: November, 2020
  - Includes support for Cyclone IV (the FPGA in our DE2-115)
  - The **Lite Edition** does ***not*** require a License Key
  - https://www.intel.com/content/www/us/en/software-kit/660907/intel-quartus-prime-lite-edition-design-software-version-20-1-1-for-windows.html
  - Downloaded file is a 5.9GB tarball (.tar file)
    - Programs (such as 7-Zip) will allow you to extract the files on a Windows computer

- Download and Install
  - The Quartus Design Software above
  - The Cyclone IV device support

USB Blaster Installation

- Browse to https://www.terasic.com.tw/wiki/Altera_USB_Blaster_Driver_Installation_Instructions
- For Windows 10, the driver is located in directory intelFPGA_lite -> 20.1 -> quartus -> drivers

New Material for this Week

- Finish covering **MIPS Instruction Set** slides
  - Start with the **JR Instruction Fields** slide **#51**

- Cover new slides
  - **MIPS Datapath - Single Memory - No Pipelining**
  - **MIPS Coding Snippets**
    - Some complete MIPS programs are available at:
      https://cscie95.dce.harvard.edu/fall2023/index.html#spim
    - The link above is to my **CSCI E-95: Compiler Design and Implementation** course

Coming Attractions

- **MIPS Assembly Language**
- **[Numeric Encodings (except for Floating Point)]**
- **Canonical Form, Minterms & Maxterms**
- **Dealing with Time in Combinational Circuits**
- **Hazards and Glitches**
- **Endianness**
- **Finite State Machines**

- **VHDL and using the Altera DE2-115 hardware**

Fifth Class Meeting on 10/1/2024

Fifth Class Meeting Agenda

- Questions and Comments
- Review of Current Problem Set Status
  - Problem Set 2
  - Principles of Operation Design Review Slots
  - Problem Set 3
- Administrivia
  - Midterm Exam
- MIPS Assembly Language
- Canonical Form, Minterms & Maxterms
- Dealing with Time in Combinational Circuits
- Hazards and Glitches
- Endianness
- Finite State Machines

Questions and Comments

- Section
- Last week's class
  - Gates
  - Boolean Logic
  - Flip-Flops, D Latch
  - Composite Devices: Register, Mux, Decoder, Half and Full Adder, ALU
  - Sum-of-Products
  - Laws and Theorems of Boolean Logic
  - Gray Codes & Karnaugh Maps
  - Computer Logic/Block Diagram
  - Two's Complement Integer Representation
  - MIPS Instruction Set
  - MIPS Pseudo Instructions
  - MIPS Coding Snippets
- Problem Sets 0, 1, 2, and 3
- Ed Discussion threads
- Readings
- Anything else

Problem Set 2

- **Problem Set 2** was due this past Sunday, September 29th, 2024
  - After PS2 is submitted, the course staff will meet with each student to review their computer instruction set and architecture through their Principles of Operation document

Principles of Operation Design Review Slots

- Wednesday, October 2: 6:00 PM: Michael
- Wednesday, October 2: 6:20 PM: Nathan
- Wednesday, October 2: 6:40 PM: Ben
- Wednesday, October 2: 7:00 PM: Patrick
- Wednesday, October 2: 7:20 PM: Leonid
- Wednesday, October 2: 7:40 PM: Frederic
- Wednesday, October 2: 8:00 PM: David
- Wednesday, October 2: 8:20 PM: Jonathan
- Wednesday, October 2: 8:40 PM: Pedro
- Thursday, October 3: 6:00 PM: Andrew
- Thursday, October 3: 6:20 PM: Julian
- Thursday, October 3: 6:40 PM: Aliya

Problem Set 3 Overview

- VHDL Counter
- Book Problems
- Final Program in High-level Language

- A word to the wise: **Complete PS3 before the midterm exam**
  - **VHDL will not be covered** in the midterm exam

Problem Set 3

- **Problem Set 3** will be due at midnight ET on Sunday, October 13th, 2024
  - *Except…*

Problem Set 3: VHDL Counter Portion

- **There is a new due date for the VHDL Counter (Problem 1) of PS3**
  - **The VHDL Counter portion of PS3 is now due on October 27th, 2024 *along with PS4***
  - The rest of Problem Set 3 is still due at midnight ET on Sunday, October 13th, 2024

Administrivia: Midterm Exam

- Two weeks until the midterm exam on Tuesday, October 15th, 2024
- We will talk about the Midterm Exam in class next week

New Material for this Week

- Cover new material
  - **MIPS Assembly Language**
  - **Canonical Form, Minterms & Maxterms**
  - **Dealing with Time in Combinational Circuits**
  - **Hazards and Glitches**
  - **Endianness**
  - **Introduction to Finite State Machines**

Coming Attractions

- **VHDL** and using the Altera DE2-115 hardware
- Implementing an assembler
- MIF file format
- Memory-mapped I/O
- Interfacing to our memory subsystem

Sixth Class Meeting on 10/8/2024

Sixth Class Meeting Agenda

- Questions and Comments
- Review of Current Problem Set Status
  - Present Problem Set 4
- Administrivia
  - Midterm Exam
  - Upcoming Dates
- Midterm Exam Information
- Finite State Machines
- VHDL and using the Altera DE2-115 hardware

Questions and Comments

- Section
- Last week's class
  - Two's Complement Integer Representation
  - MIPS Instruction Set
  - MIPS Pseudo Instructions
  - MIPS Coding Snippets
  - MIPS Assembly Language
  - Canonical Form, Minterms & Maxterms
  - Dealing with Time in Combinational Circuits
  - Timing Waveform Diagrams
  - Hazards and Glitches
  - Endianness
- Problem Sets 0, 1, 2, and 3
- Ed Discussion threads
  - ALU Design
- Readings
- Anything else

Problem Sets

- **Problem Set 3** will be due at midnight ET on this coming Sunday, October 13th, 2024
  - Except that the VHDL Counter portion of PS3 (problem 1) **is now due on October 27th, 2024 *along with PS4***
  - The rest of Problem Set 3 is still due at midnight ET on Sunday, October 13th, 2024

Problem Set 3 Overview

- VHDL Counter
- Book Problems
- Final Program in High-level Language

- A word to the wise: **Complete PS3 before the midterm exam**
  - **VHDL will not be covered** in the midterm exam

Problem Set 3: Final Program in High-level Language

- There has been an ongoing discussion in Ed about error checking of the integral inputs in the final program in high-level language
- It is quite difficult to correctly ensure that integral inputs are completely valid
  - Checking for overflow of positive and negative values in a limited word size computer (e.g. checking on a 16-bit computer that integers are not an excessively large positive value or are not an excessively bit negative value)

- Therefore, we will award extra credit for all overflow checking, but will not require any overflow checking
- Checking for valid character, etc. is still required

Present Problem Set 4

- Assembler
- Final program in your assembly language

- Show the Altera Memory Initialization File (.mif) format
  - https://cscie93.dce.harvard.edu/fall2024/def_mif.htm
  - For the DE2-115, the value for the DEPTH parameter must be 32768, the WIDTH parameter must be 16
  - These constraints are necessary because the memory system we are providing allows only the low 64K bytes of memory to be initialized in the DE2-115

- Assembly language features
  - https://cscie93.dce.harvard.edu/fall2024/slides/Assembler%20Concepts.txt

Memory-mapped I/O Interface

- https://cscie93.dce.harvard.edu/fall2024/io_interface.txt

PS4: Final Program in Assembly Language

- Parts 2 through 9 of PS4 detail subroutines to be written in your assembly language to accomplish the required final program
- All programs must be acceptable to your assembler and produce a correct .mif file
- Unfortunately, you will not be able to run and test your final program under emulation until PS5 is complete

Midterm Exam

- The midterm exam can be taken beginning next week on Tuesday, October 15th, 2024 at 8 PM ET

- Section will be held before the midterm exam, but will not deal with midterm exam material

Midterm Exam & Section Next Week

- In order to have a consistent experience for all students and because all of the class slides for use during the exam are available in Canvas, the midterm exam will be available only on-line through Proctorio
- This means that we will ***not*** be holding a midterm exam in person next week

- Next week's section will also be held remotely rather than in 53 Church Street

Upcoming Dates (1 of 2)

- Thursday, October 10th at 7:00 PM ET: Office Hours
- Sunday, October 13th at Midnight ET: Problem Set 3 is due (except for VHDL)
- Monday, October 14th at 6:30 PM ET: Office Hours
- Tuesday, October 15th at 6:45 PM ET: Section
- Tuesday, October 15th starting at 8:00 PM ET: Midterm Exam is available
- Wednesday, October 16th at 7:59 PM ET: Last time to start Midterm Exam
- Wednesday, October 16th at 11:00 PM ET: Midterm Exam is no longer available
- Thursday, October 17th at 7:00 PM ET: Office Hours

Upcoming Dates (2 of 2)

- Sunday, October 27th at Midnight ET: Problem Set 3's VHDL question is due
- Sunday, October 27th at Midnight ET: Problem Set 4 is due

Midterm Exam

- Our **Midterm Exam begins next week** on Tuesday, October 15th, 2024 at 8:00 PM ET
  - We'll talk about the midterm exam shortly
  - There will be **no class meeting on October 15th**, but **section will still be held on October 15th** at its usual time
    - No topics relevant to the midterm exam will be discussed in section

- Ask any questions relevant to the midterm exam today, in office hours, or on Ed before noon ET on the day of the exam

Midterm Exam

- Exam is **three hours in duration**
  - The exam is designed to take two hours
  - But, the exam is time-consuming – don't worry if you take all three hours
  - The problems are scored in points where a point is weighted to be approximately one minute of exam answer duration
- Earliest starting time is 8:00 PM ET on Tuesday, October 15th, 2024 – **one week from today**
- Latest starting time is 7:59 PM ET on Wednesday, October 16th, 2024
- The exam will be available through **Canvas** under the **Quizzes** tab
  - The exam will be administered under **Proctorio**
  - **Google Chrome, Microsoft Edge, Brave, or Opera must be used as the browser** for the exam – this is a Proctorio requirement
  - Proctorio requires that a **Proctorio Extension** be installed into your browser
- No books may be used during the exam, but all slides from class are available through the **Files** menu item in Canvas
  - Nothing else can be used during the exam: no notes, no electronics other than the computer being used for the exam, no cell phones, no communication
  - But, nothing else is needed!

The Proctorio Setup Quiz

- *As soon as possible,* take the **Proctorio Setup Quiz** that is available in **Canvas** under **Quizzes**
- Really **carefully read all of the instructions**
  - **The instructions are the same as for the real Midterm Exam**
  - **We are not able to deal with issues caused by not following the instructions!**
- Contact the course staff *as soon as possible* if you have any problems taking the **Proctorio Setup Quiz**

Midterm Exam Format

- Similar to book problems
- Hardware design
- Multiple choice question(s)
  - Choose the correct answer(s)
- Some questions with short answers
  - Text box in which answer can be typed
- Some questions that require drawings as answers
  - Create a drawing and upload it to the Canvas quiz site

- A significant problem that requires thinking outside the box and is new and relevant for later in the class, but based on what we have already covered

Midterm Exam Material

- All material covered in class through – and including – today's class meeting and on the Class Website
- Material covered in Problem Sets 0-3 (except for VHDL)
- All readings on the syllabus through and including today's readings
- Discussions on Ed
- Material covered in section

- No **VHDL** will be on the exam

Slide Deck Material that is Midterm Eligible

- **Binary Logic Levels**
- **Boolean Logic**
- **Boolean Logic Continued**
- **Advanced Boolean Logic**
- **Laws and Theorems of Boolean Logic**
- **Computer Logic**
- **Place Values**
- **Numeric Encodings (Two's Complement only)**
- **Gray Codes & Karnaugh Maps**
- **Canonical Form, Minterms & Maxterms**
- **Dealing with Time in Combinational Circuits**
- **MIPS Instruction Set**
- **MIPS Datapath - Single Memory - No Pipelining**
- **MIPS Coding Snippets**
- **MIPS Assembly Language**
- **Hazards and Glitches**
- **Endianness**
- **Finite State Machines**

Midterm Directions

- Read each question very carefully
  - The question should include all necessary information

- Present any truth tables in the format and order requested

- The point values in the exam are equal to the number of minutes that a student who would score highly might take to complete the question
- The exam in lengthy – don’t be concerned if you don’t complete everything
- Initially, spend approximately the number of minutes on each question that the question is worth, then return to the question later to complete them as available time permits

New Material for this Week

- Cover new material
  - **Finite State Machines (FSM**s**)**

- Later today
  - **VHDL** and using the Altera DE2-115 hardware

Application Notes

- For DE2-115, if you see the following warning, take a look at AN1
  - Warning (15714): Some pins have incomplete I/O assignments. Refer to the I/O Assignment Warnings report for details

  - Also, for information about other issues, download and view the DE2-115 FAQ from Terasic which is available at: https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=165&No=502&PartNo=4

Encountered a Problem Installing the USB Blaster Driver?

- See http://www.terasic.com.tw/wiki/Windows_encountered_a_problem_installing_the_drivers_for_your_device

Show the DE2-115 User Manual

- The DE2-115 User Manual is available from Terasic at:
  - https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=165&No=502&PartNo=4

- Do not program the Flash memory
  - That is, ensure that the **slide switch is always in the RUN position** (not in the PROG position)

- **Slide (Toggle) Switches**: Section 4.2 on Pages 32-34 and Table 4-1 on Page 35
- **Pushbutton Switches**: Section 4.2 on Pages 32-33 and Table 4-2 on Page 35
- **LEDs**: Section 4.3 on Page 34 and Table 4-3 on Pages 35-36
- **Seven-segment Displays**: Section 4.4 on Page 36 and Table 4-4 on Pages 36-38
- **Clocks**: Section 4.5 on Page 38 and Table 4-5 on Page 38

VHDL and Using the DE2 Hardware

- Cover new material
  - **VHDL**
    - Cover through the **Concurrent Assignment Ordering** slide

Coming Attractions

- More **VHDL**
- Implementing an assembler
- **Numeric Encodings** (other than two's complement)

Seventh Class Meeting on 10/15/2024

- Midterm Exam

Eighth Class Meeting on 10/22/2024

Eighth Class Meeting Agenda

- Welcome back! And, long time, no see!
- Questions and Comments
- Review of Current Problem Set Status
  - Status of Coursework
  - Present Preliminary Final Project Problem Set (ALU)
- Administrivia
  - Midterm Exam Grading
- Intel/Altera DE2-115 Hardware
- Problem Set 4
  - Emitted MIF File
  - Assembly Code
  - Assembler Approach
- Quick introduction to edge-triggered memory devices
- Continue with VHDL and using the Altera DE2-115 hardware

Questions and Comments

- Section
- Midterm exam
- Earlier class meetings
  - Your instruction set/assembly language
  - C program for required final demonstration
  - Translating your C program into your assembly language
  - Assembler implementation
  - Memory-mapped I/O
  - Altera Memory Initialization File (.mif) format
  - VHDL
- Problem Sets 0 through 4
- Ed Discussion threads
- Readings
- Anything else

Status of Coursework

- This is the point in this course where students have the highest workload
- This is also the point where students may feel that the path from where we are now to completion of the course is uncertain and excessively difficult
  - The course staff will lead you through the steps to final project completion
  - The upcoming problem sets are more straight-forward and less time-consuming
- However, it is imperative that you contact the course staff to resolve any questions that you have

Problem Sets

- **Problem Set 3** *except for the VHDL Counter portion* was due at midnight ET on Sunday, October 13th, 2024

- The **VHDL Counter portion of Problem Set 3** and all of **Problem Set 4** will be due at midnight ET on this coming Sunday, October 27th, 2024

- The **Preliminary Final Project Problem Set (ALU)** will be due at midnight ET on Sunday, November 3rd, 2024

Review the VHDL Counter Problem in PS3

- Reminders
  - Use KEY3 as an asynchronous clear
  - Use KEY2 as the clock to count up
  - Don't try to debounce pushbutton KEY2
    - (Any contact bounce from KEY3 is irrelevant and idempotent)
  - The counter is an 8-bit counter
    - Therefore, incrementing from 0xff will yield 0x00
    - No special action is required for this to occur
- Hint
  - Look at code on the class website
    - 4-bit register using std_ulogic with **synchronous** clear: regstdulogicwithsyncclear.vhd
    - 4-bit register using std_ulogic with **asynchronous** clear: regstdulogicwithasyncclear.vhd
    - 4-bit register using std_ulogic with **synchronous** clear that has two entities in a single .vhd file and shows how to implement a mapping: regstdulogicwithsyncclearmappingwith2entities.vhd
- For this problem set, just put all of your entities into a single .vhd file

Present the Preliminary Final Project Problem Set (ALU)

- ALU for your CPU in VHDL
- Must be based on a bit-slice ALU component

- Should be implemented using combinational logic
- In VHDL for this assignment, a process should **not** be required

State of Midterm Exam Grading

- We’re not finished grading the midterm exam
- We hope to review the Midterm Exam solutions next week

Intel/Altera DE2-115 Hardware

- It is imperative that everyone has built and loaded projects onto the Intel/Altera DE2-115 hardware and tested that they function correctly
- This demonstrates that Quartus, the driver, and the hardware are working correctly

Emitted MIF File

- For the DE2-115,
  - DEPTH parameter must be 32768
  - WIDTH parameter must be 16
- Both the ADDRESS_RADIX and the DATA_RADIX should be HEX
- Note that in a MIF file, the hex numbers do **not** have a 0x prefix
- Remember that our memory subsystem is **little-endian**
- Also, the address before the colon is the address of a 16-bit word
  - So, even though your computer may be byte addressable, ***in the MIF file***, the addresses are for 16-bit words
  - Here is an example of a MIF file and the comments detail how the contents would appear to a byte-addressable computer

```
0000: 0001          -- Address 0000 contains 01; address 0001 contains 00
0001: BB0A          -- Address 0002 contains 0A; address 0003 contains BB
000F: 1DEA          -- Address 001E contains EA; address 001F contains 1D
```

Problem Set 4 Assembly Code

- Issues
  - How to do I/O?
    - Character-by-character at the lowest level of your assembly code
    - Need to check the appropriate ready bit for input & output

Problem Set 4 Assembler Approach

- Issues
  - What is a symbol table?
  - How to build the symbol table
    - Two passes
    - Other approaches: One pass, Multiple passes
    - Assume addresses start at 0 at the beginning of the assembly code input file
      - Increment the assembler's location counter (address) as progress is made through the assembly code input file, but don't emit any code or data
      - Add all labels with their associated addresses into the symbol table in the first pass
  - Data vs. Instructions
  - Emit the MIF
    - Output MIF as the second pass is made
    - Two approaches
      - Store all translated data & instructions as unsigned integers in an array and then emit to MIF after the second pass is complete
      - Or, emit to MIF as each line of assembly input is processed during the second pass

New Material for this Week

- Cover new material
  - Continue covering **VHDL** slides
    - Start with the **Processes** slide #31
    - Cover **VHDL** through the **Using VHDL to Implement an FSM (5 of 5)** slide

Ninth Class Meeting on 10/29/2024

Ninth Class Meeting Agenda

- Questions and Comments
- Administrivia
- Grades and Coursework
- Overall Schedule
- Coming Attractions
- Review of Current Problem Set Status
  - Status of Coursework
  - Present Problem Set 5
- Review the Midterm Exam
- Continue with the **VHDL** slides
- Present the **VHDL Looping** slides
- Quick introduction to edge-triggered memory devices
  - See Edge-Triggered Flip-Flops slides
- Show the new VHDL project memorySubsystemFrameworkDE2_115
  - Memory subsystem integration
  - How to load MIF file
  - How to examine memory

Questions and Comments

- Section
- Midterm exam
  - Please delay asking any questions relevant to the midterm exam until we review the exam answers later today
- Earlier class meetings
  - Your instruction set/assembly language
  - C program for required final demonstration
  - Translating your C program into your assembly language
  - Assembler implementation
    - Dealing with labels and the symbol table
    - Dealing with immediate fields for branches and constants larger than the immediate field size
  - Memory-mapped I/O
  - Altera Memory Initialization File (.mif) format
  - VHDL
    - Finite State Machine implemented in VHDL
- Problem Sets 0 through 4
  - Counter in VHDL
- Preliminary Final Project problem set
- Ed Discussion threads
- Readings
- Anything else

Administrivia (1 of 2)

- Our class ski trip will take place on Sunday, January 26th, 2025 at Killington Ski Resort, Vermont
- Please reserve the date!
- We'll furnish more information as the semester progresses

Administrivia (2 of 2)

- Have a fun Halloween!

Grades and Coursework (1 of 2)

- If you're concerned about your grades, please talk with the course staff before taking any action

- We understand that much of our current coursework
  - requires students to create a new computer model
  - deals with new concepts, new abstractions, and new languages
  - requires iterative design work causing revisions to prior assignments
  - requires that each new problem set is built upon previous problem sets

- Therefore…

Grades and Coursework (2 of 2)

- **We will *not* be deducting late points from any Problem Set grades**

- However, you will still need to complete your final project; so, keep on track

Overall Schedule

- To complete your final project, it is important for you to be up-to-date with a working assembler, assembly language programs, and emulator, and a finalized ISA documented in a current version of your Principles of Operation manual by the due date for PS5 (Sunday, November 10, 2024)

Coming Attractions

- Pipelining
- Shifters
- PDP-8 instruction set
- PDP-11 instruction set

Problem Sets

- The **VHDL Counter portion of Problem Set 3** and all of **Problem Set 4 (Assembler** and **Assembly Language Programs)** was due at midnight ET on this past Sunday, October 27th, 2024

- The **Preliminary Final Project Problem Set (ALU in VHDL)** will be due at midnight ET on this coming Sunday, November 3rd, 2024

- **Problem Set 5 (Emulator)** will be due at midnight ET on Sunday, November 10th, 2024

Present Problem Set 5

- Emulator/Simulator

- Now, you'll finally be able to test your code written for PS4!
- Your emulator may also serve as a software model for the correct implementation of instructions in your CPU

- Corrections/modifications to:
  - ISA (Instruction Set Architecture)
  - Block Diagram
  - Principles of Operation documentation
  - Final program in your assembly language
  - Assembler

Emulator Implementation (1 of 3)

- Your emulator **need not** implement your architecture in the same way that your hardware/VHDL/FPGA will implement your architecture
  - Architecturally identical
    - All aspects that an assembly language programmer could see and all aspects that are described in your Principles of Operation documentation
  - Not organizationally identical
    - Does not need to implement instructions/sequencer in the same way that the hardware will implement these aspects
- Our memory subsystem is **little endian**

Emulator Implementation (2 of 3)

- You must implement I/O to the serial (RS-232) port addresses
  - **You will not be using an actual RS-232 port**
  - The emulated RS-232 I/O will be from stdin and to stdout
- Your emulator **need not** implement the PS/2 and LCD ports and their status/control bits

- It is acceptable to assume that serial (RS-232) input and output are always ready
- If you implement this approach, any attempt to read a character will stall until a character is typed (if the input is buffered as a line, the input operation will stall until a complete line is typed)
- Extra credit will be awarded for simulating the input ready bit

Emulator Implementation (3 of 3)

- You must implement some form of tracing
  - Minimally: display the PC of each emulated instruction before executing it

- A multitude of extra credit options:
  - Display all registers before executing each instruction
  - Display only those registers that were modified by the previous instruction before executing an instruction
  - Update registers in a separate window pane or area
    - Use a different color to indicate modification
  - Display of disassembly (or assembly source code) of each instruction ***before*** execution
  - Single stepping
  - Breakpoints
  - Watch points for memory data locations
  - …

State of Midterm Exam Grading

- We've finished grading the midterm exams and comments and grades are available through **Canvas -> Quizzes**
  - **Comments are in the Additional Comments** box after each question & answer

Summary of Midterm Grades

Name	Midterm 1	Midterm 2	Midterm 3	Midterm 4	Midterm 5	Midterm 6	Midterm 7	Midterm 8	Midterm 9	Midterm 10	Midterm 11	Midterm Tot
	Integer Repre	Integer Repre	Timing Wavef	Timing Wavef	Timing Wavef	Timing Wavef	MIPS	Laws and The	Boolean Sim	ALU Design	Mealy Finite	State Machine
Max Possible	2	3	5	10	5	5	5	4	6	30	45	120
Count	10	10	9	9	9	10	9	10	10	9	10	10
Minimum	0	0	1	0	3	3	2	1	1	0	5	12
Maximum	2	3	5	10	5	5	5	4	6	28	40	107
Average	1.4	2.5	4	7.22222222	4.22222222	4.1	3.77777778	2.7	4.4	17.7777778	23.9	72.3
St. Dev.	0.6992059	1.08012345	1.32287566	2.99072641	0.83333333	0.73786479	1.48136574	1.05934991	1.83787317	9.39118972	10.8161402	26.5917698
Median	1.5	3	4	8	4	4	5	2.5	5	21	25.5	74

Present Solutions to Midterm Exam

Tenth Class Meeting on 11/5/2024

Tenth Class Meeting Agenda

- Questions and Comments
- Administrivia
- Overall Schedule
- Coming Attractions
- Review of Current Problem Set Status
  - Status of Coursework
- I'm delaying presenting Problem Set 6 until next week
  - I'll also delay presenting the memory interaction protocol until next week
- Continue with the **VHDL** slides
- Present the **VHDL Looping** slides
- Quick introduction to edge-triggered memory devices
  - See Edge-Triggered Flip-Flops slides
- Show the new VHDL project memorySubsystemFrameworkDE2_115
  - Memory subsystem integration
  - How to load MIF file
  - How to examine memory
- Final Project Clocking Scheme

Questions and Comments

- Section
- Midterm exam
- Earlier class meetings
  - Your instruction set/assembly language
  - C program for required final demonstration
  - Translating your C program into your assembly language
  - Assembler implementation
    - Dealing with labels and the symbol table
    - Dealing with immediate fields for branches and constants larger than the immediate field size
  - Memory-mapped I/O
  - Altera Memory Initialization File (.mif) format
  - VHDL
- Problem Sets 0 through 5
  - PS3 Counter in VHDL
  - PS4 Assembler, Assembly language programs
    - Dealing with labels and the symbol table
    - Dealing with immediate fields for branches and constants larger than the immediate field size
    - Performing I/O in assembly language
    - MIF file format
  - PS5 Emulator implementation
- VHDL
- Preliminary Final Project problem set
- Ed Discussion threads
- Readings
- Anything else

Administrivia (1 of 2)

- **We will *not* be deducting late points from Problem Set grades**

- However, you will still need to complete your final project; so, keep on track
- To complete your final project, it is important for you to be up-to-date with a working assembler, assembly language programs, and emulator, and a finalized ISA documented in a current version of your Principles of Operation manual by the due date for PS5 (Sunday, November 10, 2024)

Administrivia (2 of 2)

- Our class ski trip will take place on Sunday, January 26th, 2025 at Killington Ski Resort, Vermont
- Please reserve the date!
- We'll furnish more information as the semester progresses

Overall Schedule

- To complete your final project, it is important for you to be up-to-date with a working assembler, assembly language programs, and emulator, and a finalized ISA documented in a current version of your Principles of Operation manual by the due date for PS5 (Sunday, November 10, 2024)

Coming Attractions

- PS6 description
- Pipelining
- PDP-8 instruction set
- PDP-11 instruction set

Problem Sets

- The **VHDL Counter portion of Problem Set 3** and all of **Problem Set 4 (Assembler and Assembly Language Programs)** was due at midnight ET on Sunday, October 27th, 2024
- The **Preliminary Final Project Problem Set (ALU in VHDL)** was due at midnight ET on this past Sunday, November 3rd, 2024
- **Problem Set 5 (Emulator)** will be due at midnight ET on Sunday, November 10th, 2024
- **Problem Set 6 (Sequencer action description, VHDL memory system interaction)** will be due at midnight ET on Sunday, November 24th, 2024
  - We'll go over material relevant to this problem set in next week's class

Preliminary Final Project Problem Set (ALU)

- ALU for your CPU in VHDL
- Must be based on a bit-slice ALU component

- Should be implemented using combinational logic
- In VHDL, a process should **not** be required
  - You're allowed to use a process if you want to use VHDL constructs that must be inside a process; however, a process is not needed to complete the VHDL code (*i.e.,* the ALU should be strictly combinational logic)
  - Remember to use std_ulogic throughout

Problem Set 5

- Emulator/Simulator

- You do ***not*** need to be able to parse the entire MIF specification; you only need to be able to read any MIF file emitted by your assembler

- Now, you'll finally be able to test your code written for PS4!
- Your emulator may also serve as a software model for the correct implementation of instructions in your CPU

- Corrections/modifications to:
  - ISA (Instruction Set Architecture)
  - Block Diagram
  - Principles of Operation documentation
  - Final program in your assembly language
  - Assembler

New Material for this Week

- Cover new material
  - Continue covering **VHDL** slides
    - Start with the **Using Configuration to Select One of Several Architectures** slide #87
    - Deep dive into the FSM implemented in VHDL
  - Present the **VHDL Looping** slides
  - Present the **Edge-Triggered Flip Flop** slides

New VHDL project: memorySubsystemFrameworkDE2_115

- Show the new VHDL project memorySubsystemFrameworkDE2_115
  - Memory subsystem integration
  - How to load MIF file
  - How to examine memory

Some Memory Subsystem Requirements

- **DO NOT SET** THE LEAST SIGNIFICANT BIT of the **clock_divide_limit**
- Device and Pin Options: Reserve all unused pins: As input tri-stated
- Read the documentation for the memory subsystem that needs to be compiled with student projects
- **cscie93.sdc**: this file specifies some clock and timing information for the Altera TimeQuest Timing Analyzer
    - To enable an accurate timing analysis, and avoid warnings about undefined clocks, you should include this file in your project as well
    - Follow the directions in the Building a Project with the Memory Subsystem slides
- Under **Assignments -> Settings…** continuing to **Compiler Settings** -> **Advanced Settings (Fitter)…** , change **Fitter Effort** to **Standard Fit**

VHDL

- Answer any questions about the VHDL programs on the class website at **Hardware Related References** -> **VHDL Programs**
  - 4-bit comparator versions
  - Invert segment versions
  - Using FSMs:
    - Debounce switch
    - Rotate segments
  - Identify segments
  - Up-down counter

- storeSwitchValueUsingFSM
  - Uses an FSM for state transitions and sets control lines based on the state of the FSM – thus, it is a ***Moore*** FSM
  - See https://cscie93.dce.harvard.edu/fall2024/slides/Moore FSM for Store Switch Value.pdf (https://cscie93.dce.harvard.edu/fall2024/slides/Moore FSM for Store Switch Value.pptx)
  - **This is the same procedure that we want you to use for your final project FSM(s)**

Additional New Material for this Week

- Cover new material
  - **Final Project Clocking Scheme** briefly reviewed

Eleventh Class Meeting on 11/12/2024

Eleventh Class Meeting Agenda

- Questions and Comments
- Administrivia
- Overall Schedule
- Coming Attractions
- Review of Current Problem Set Status
  - Status of Coursework
  - Present Problem Set 6
  - **Final Project Clocking Scheme** slides
- Present the **memory interaction protocol**
- Shifters
- Comparators
- Pipelining

Questions and Comments

- Section
- Midterm exam
- Earlier class meetings
  - Your instruction set/assembly language
  - C program for required final demonstration
  - Translating your C program into your assembly language
  - Assembler implementation
    - Dealing with labels and the symbol table
    - Dealing with immediate fields for branches and constants larger than the immediate field size
  - Memory-mapped I/O
  - Altera Memory Initialization File (.mif) format
  - VHDL
  - Clocking Scheme
  - Shifters
  - Comparators
- Problem Sets 0 through 5
  - PS3 Counter in VHDL
  - PS4 Assembler, Assembly language programs
    - Dealing with labels and the symbol table
    - Dealing with immediate fields for branches and constants larger than the immediate field size
    - Performing memory-mapped I/O in assembly language
    - MIF file format
  - PS5 Emulator implementation
- VHDL
- Preliminary Final Project problem set
- Ed Discussion threads
- Readings
- Anything else

Administrivia (1 of 2)

- **We will *not* be deducting late points from Problem Set grades**

- However, you will still need to complete your final project; so, keep on track
- To complete your final project, it is important for you to be up-to-date with a working assembler, assembly language programs, and emulator, and a finalized ISA documented in a current version of your Principles of Operation manual by approximately this week

- If you have any questions about your status in the course, please reach out to the course staff

Administrivia (2 of 2)

- Our class ski trip will take place on Sunday, January 26th, 2025 at Killington Ski Resort, Vermont
- Please reserve the date!
- We'll furnish more information as the semester progresses

Coming Attractions

- **Register Array** slides
- Present AN5 & AN7 on using the DE2-115 serial port
- Floating-Point Number Representation from the **Numeric Encodings** slides
- **PDP-8 instruction set** slides
- **PDP-11 instruction set** slides
- **Caching** slides
- **Virtual Memory** slides

Problem Sets

- The **VHDL Counter portion of Problem Set 3** and all of **Problem Set 4 (Assembler and Assembly Language Programs)** was due at midnight ET on Sunday, October 27th, 2024
- The **Preliminary Final Project Problem Set (ALU in VHDL)** was due at midnight ET on Sunday, November 3rd, 2024
- **Problem Set 5 (Emulator)** was due at midnight ET on this past Sunday, November 10th, 2024
- **Problem Set 6 (Sequencer action description, VHDL memory system interaction)** will be due at midnight ET on Sunday, November 24th, 2024
  - We'll go over material relevant to this problem set in class today

Preliminary Final Project Problem Set (ALU)

- ALU for your CPU in VHDL
- Must be based on a bit-slice ALU component

- Should be implemented using combinational logic
- In VHDL, a process should **not** be required
  - You're allowed to use a process if you want to use VHDL constructs that must be inside a process; however, a process is not needed to complete the VHDL code (*i.e.,* the ALU should be strictly combinational logic)
  - Remember to use std_ulogic throughout

Problem Set 5

- Emulator/Simulator

- You do ***not*** need to be able to parse the entire MIF specification; you only need to be able to read any MIF file emitted by your assembler

- Now, you'll finally be able to test your code written for PS4!
- Your emulator may also serve as a software model for the correct implementation of instructions in your CPU

- Corrections/modifications to:
  - ISA (Instruction Set Architecture)
  - Block Diagram
  - Principles of Operation documentation
  - Final program in your assembly language
  - Assembler

Problem Set 6

- Problem 1 (Counts toward Problem Set grade): Detailed description of what happens on each cycle and for each clock edge for all instructions
  - Shortly, I'll present the memory interaction protocol from https://cscie93.dce.harvard.edu/fall2024/processor_memory_interface.txt
  - Your solution to this problem will be presented as a spreadsheet as will be shown in class now
    - Present VHDL for PS3, for ALU Testbench, and for PS6 Memory Interaction under Slides used in class
    - Rows are states
    - Columns are next states and control lines
      - Next state may be conditionalized based on IR contents (opcode, function code) & status lines, etc.
  - Multi-bit control lines (for example, for a four-way MUX) can be listed as a single control line with each enumerated-type value selecting a different input

- Problem 2 (Counts toward Final Project grade): VHDL code to interact with the memory subsystem – read followed by write in an infinite loop
  - To be implemented as a **Moore** Finite State Machine using the same overall design as the storeSwitchValueUsingFSM.vhd VHDL code on the class website

PS6: Problem 1

- Explore the **Final Project Clocking Scheme** slides in detail

- Reminder of how clocking will work in the final project
  - Clock goes everywhere always
  - Enables are turned on by the controller FSM

- Clocking scheme
  - Progress from one FSM state to another on clock falling edge
  - While in an FSM state (*i.e.,* when the clock is low), assert appropriate enables and control lines (*e.g.,* ALU function, MUX control)
  - While clock is low, signals propagate through the data paths
  - Clock all registers on clock rising edge
    - Only those registers that are *enabled* will be updated
    - This maintains a consistent point in time at which all registers are simultaneously updated

Some Memory Subsystem Requirements

- **DO NOT SET** THE LEAST SIGNIFICANT BIT of the **clock_divide_limit**
- Device and Pin Options: Reserve all unused pins: As input tri-stated
- Read the documentation for the memory subsystem that needs to be compiled with student projects
- **cscie93.sdc**: this file specifies some clock and timing information for the Altera TimeQuest Timing Analyzer
    - To enable an accurate timing analysis, and avoid warnings about undefined clocks, you should include this file in your project as well
    - Follow the directions in the Building a Project with the Memory Subsystem slides
- Under **Assignments -> Settings…** continuing to **Compiler Settings** -> **Advanced Settings (Fitter)…** , change **Fitter Effort** to **Standard Fit**

PS6: Problem 2 (1 of 2)

- Present the memory interaction protocol from https://cscie93.dce.harvard.edu/fall2024/processor_memory_interface.txt
- Additional signals are shown in https://cscie93.dce.harvard.edu/fall2024/MemorySubsystemSummary%2020141118.pdf
  - These include:
    - clock_hold – stops sysclk1 & sysclk2
    - clock_step – if clock_hold is already asserted, a rising edge on clock_step will generate a single full clock cycle
    - clock_divide_limit – a 20-bit std_logic_vector to slow sysclk1 & sysclk2
    - sysclk1 – use this clock for your processor
    - sysclk2 – 180° out of phase from sysclk1; can be used for your processor
    - serial_character_ready – used only if the CPU implements a hardware serial port interrupt
    - ps2_character_ready – used only if the CPU implements a hardware PS/2 port interrupt
  - In addition, there are many signals that need to be connected directly to pins

PS6: Problem 2 (2 of 2)

- The memory subsystem VHDL code is available on the class web site

- For DE2-115:
  - https://cscie93.dce.harvard.edu/fall2024/tools/Memory%20Subsystem%20VHDL%2020141118%20cscie93_DE2-115.zip
- See the DE2-115 Top-Level Shell File in the Memory Subsystem Summary document above and also in the program that instantiates the memory subsystem and allows loading and validation of MIF files: memorySubsystemFrameworkDE2_115.vhd

New Material for this Week

- Cover new material
  - **Shifters**
  - **Comparators**

Additional New Material for this Week

- Cover new material
  - **Pipelining** through the **Pipelining** slide #3
    - Including the discussion of pipelining as demonstrated by a Harvard Square taqueria

Twelfth Class Meeting on 11/19/2024

Twelfth Class Meeting Agenda

- Questions and Comments
- Administrivia
- Coming Attractions
- Review of Current Problem Set Status
- Final Project Presentations
- VHDL/FPGA reminders
- Memory subsystem requirements
  - Dealing with mem_reset
- Present Application Notes
  - Present AN3, AN4 & AN8 on using RAM in Quartus
  - Present AN6 on building a Quartus project that uses the memory subsystem
- Cover new material
  - Continue covering **Pipelining** slides
    - Start with the **Pipelining** slide #3
  - **Register Array** slides
  - **Caching** slides

Questions and Comments

- Section
- Midterm exam

- Earlier class meetings
  - Your instruction set/assembly language
  - C program for required final demonstration
  - Translating your C program into your assembly language
  - Assembler implementation
    - Dealing with labels and the symbol table
    - Dealing with immediate fields for branches and constants larger than the immediate field size
  - Memory-mapped I/O
  - Altera Memory Initialization File (.mif) format
  - VHDL
  - Clocking Scheme
  - Shifters
  - Comparators
  - Pipelining
  - VHDL

- Problem Sets 0 through 6
  - PS3 Counter in VHDL
  - PS4 Assembler, Assembly language programs
    - Dealing with labels and the symbol table
    - Dealing with immediate fields for branches and constants larger than the immediate field size
    - Performing memory-mapped I/O in assembly language
    - MIF file format
  - PS5 Emulator implementation
  - PS6 Interfacing with the memory subsystem
    - How to display and set memory contents from inside Quartus – this is limited to the memory accessible in the MIF
  - PS6 Spreadsheet of what happens on each cycle and for each clock edge in sequencer for all instructions

- Preliminary Final Project problem set
- Ed Discussion threads
- Readings

- Anything else

Administrivia (1 of 3)

- **We will *not* be deducting late points from Problem Set grades**

- However, you will still need to complete your final project; so, keep on track
- To complete your final project, it is important for you to be up-to-date with a working assembler, assembly language programs, and emulator, and a finalized ISA documented in a current version of your Principles of Operation manual around now
- Your ALU in VHDL and VHDL code to read then write to the memory subsystem need to be working

- If you have any questions about your status in the course, please reach out to the course staff

Administrivia (2 of 3)

- We will be holding section and class meetings next week on Tuesday, November 26th, 2024

- The Extension School Thanksgiving Break begins on Wednesday, November 27th, 2024

Administrivia (3 of 3)

- Our class ski trip will take place on Sunday, January 26th, 2025 at Killington Ski Resort, Vermont
- Please reserve the date!
- We'll furnish more information as the semester progresses

Coming Attractions

- **Hardware Multiplication** slides
- **Serial Communication** slides
- **PDP-8 instruction set** slides
  - **Opcode 6** is for **In-Out Transfer** instructions
- **PDP-11 instruction set**
  - **PDP-11 Handbook**
  - **PDP-11 Architecture Handbook** through Register Deferred Mode addressing mode
- Present AN5 & AN7 on using the DE2-115 serial port
- Floating-Point Number Representation from the **Numeric Encodings** slides
- **Virtual Memory** slides

Problem Sets

- The **VHDL Counter portion of Problem Set 3** and all of **Problem Set 4 (Assembler and Assembly Language Programs)** was due at midnight ET on Sunday, October 27th, 2024

- The **Preliminary Final Project Problem Set (ALU in VHDL)** was due at midnight ET on Sunday, November 3rd, 2024

- **Problem Set 5 (Emulator)** was due at midnight ET on Sunday, November 10th, 2024

- **Problem Set 6 (Sequencer action description, VHDL memory system interaction)** will be due at midnight ET on this coming Sunday, November 24th, 2024

Problem Set 6

- Problem 1 (Counts toward Problem Set grade): Detailed description of what happens on each cycle and for each clock edge for all instructions
  - Memory interaction protocol described in https://cscie93.dce.harvard.edu/fall2024/processor_memory_interface.txt
  - Your solution must be presented as a spreadsheet as was shown in class last week
    - See VHDL for PS3, for ALU Testbench, and for PS6 Memory Interaction under Slides used in class
    - Rows are states
    - Columns are next states and control lines
      - Next state may be conditionalized based on IR contents (opcode, function code) & status lines, etc.
      - A conditional next state can be written as an expression (*e.g.,* if IR[15..12]=X"5" and IR[3..0]=X"0" then newState:=instAdd)
  - Multi-bit control lines (for example, for a four-way MUX) can be listed as a single control line with each enumerated-type value selecting a different input

- Problem 2 (Counts toward Final Project grade): VHDL code to interact with the memory subsystem – read followed by write in an infinite loop
  - To be implemented as a **Moore** Finite State Machine using the same overall design as the storeSwitchValueUsingFSM.vhd VHDL code on the class website

Final Project Presentations

- Each student will present their final project to the class and visitors during our final class meeting on Tuesday, December 17th, 2024
- Our final class meeting will begin at 6:45 PM ET

- The project presentation meeting is just four weeks from today!

- All final project code, documentation, and presentation material **must** be submitted by **2 PM ET on Friday, December 20th, 2024**

VHDL/FPGA Reminders

- For DE2-115, if you see the following warning, take a look at AN1: Dealing with DE2-115 Current Strength and Slew Rate: AN1 DE2-115 Current Strength and Slew Rate.txt
  - Warning (15714): Some pins have incomplete I/O assignments. Refer to the I/O Assignment Warnings report for details
  - Also, for information about other issues, download and view the DE2-115 FAQ from Terasic which is available at: https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=165&No=502&PartNo=4
- Device and Pin Options: Reserve all unused pins: As input tri-stated

Memory Subsystem Requirements

- **DO NOT SET** THE LEAST SIGNIFICANT BIT of the **clock_divide_limit**
- Read the documentation for the memory subsystem that needs to be compiled with student projects
- **cscie93.sdc**: this file specifies some clock and timing information for the Altera TimeQuest Timing Analyzer
    - To enable an accurate timing analysis, and avoid warnings about undefined clocks, you should include this file in your project as well
    - Follow the directions in the Building a Project with the Memory Subsystem slides
- Under **Assignments -> Settings…** continuing to **Compiler Settings** -> **Advanced Settings (Fitter)…** , change **Fitter Effort** to **Standard Fit**

mem_reset

- When **mem_reset** is asserted to the memory subsystem, no **sysclk1** or **sysclk2** clock pulses are generated
- Therefore, your use of **mem_reset** to reset your FSM(s) to its/their initial state(s) must be asynchronous to the clock (*i.e.,* because you will not receive a clock pulse when **mem_reset** is asserted, you cannot use a usual state transition that checks for **mem_reset** to cause your FSM(s) to be in its/their initial (idle/reset) state)
- So, you should use **mem_reset** to reset your FSM(s), but check for **mem_reset** before you wait for the (falling) edge of the clock (**sysclk1**)

Sample FSM Reset VHDL Code

```
stateMachine: process(mem_reset, sysclk1) is
    variable newState: StateType;
  begin
    if mem_reset = '1' then
      newState := state_reset;
    elsif falling_edge(sysclk1) then
      case presentState is
        when state_reset =>
          newState := state_read_step_1;
        when state_read_step_1 =>
          if mem_dataready_inv = '0' then
            newState := state_read_step_1;
          else
            ...
```

VHDL Project Reminders

- Hardware Devices
  - All hardware devices (pushbuttons, slide switches, individual red and green LEDs, seven-segment displays, clocks, SSRAM, RS-232, LCD, PS/2) must appear as ports in the ***top-level entity***
- Top-level Ports
  - None of the memory subsystem top-level entity port declarations should be changed
  - **Don't declare any** of ***your top-level ports*** as **inout** or **buffer**
  - All top-level entity ports should be assigned to pins
- Remember to **have an initial reset state** in which nothing happens
  - This is because after being reset, the first clock edge is a rising edge
  - There is an example of doing this in the previous slide

Present Application Notes

- Present AN3, AN4 & AN8 on using RAM in Quartus
- Present AN6 on building a Quartus project that uses the memory subsystem

New Material for this Week

- Continue covering **Pipelining** slides
  - Start with the **Pipelining** slide #3

- Cover new material
  - **Register Array** slides
  - **Caching** slides

Thirteenth Class Meeting on 11/26/2024

Thirteenth Class Meeting Agenda

- Questions and Comments
- Administrivia
- Coming Attractions
- Review of Current Problem Set Status
- Final Project Presentations
- FPGA/VHDL Implementation & Memory subsystem requirements
- Cover new material
  - **Hardware Multiplication** slides
  - **Serial Communication** slides
  - **PDP-8 instruction set** slides
    - **Opcode 6** is for **In-Out Transfer** instructions
  - **PDP-11 instruction set**
    - **PDP-11 Handbook**
    - **PDP-11 Architecture Handbook**
  - Present AN5 & AN7 on using the DE2-115 serial port
  - Floating-Point Number Representation from the **Numeric Encodings** slides

Questions and Comments

- Section
- Midterm exam

- Earlier class meetings
  - Your instruction set/assembly language
  - C program for required final demonstration
  - Translating your C program into your assembly language
  - Assembler implementation
    - Dealing with labels and the symbol table
    - Dealing with immediate fields for branches and constants larger than the immediate field size
  - Memory-mapped I/O
  - Altera Memory Initialization File (.mif) format
  - VHDL
  - Clocking Scheme
  - Shifters
  - Comparators
  - Pipelining
  - VHDL
  - Edge-triggered flip-flops
  - Register array
  - Serial communication
  - PDP-8 or introduction to PDP-11 instruction set

- Problem Sets 0 through 6
  - PS3 Counter in VHDL
  - PS4 Assembler, Assembly language programs
    - Dealing with labels and the symbol table
    - Dealing with immediate fields for branches and constants larger than the immediate field size
    - Performing memory-mapped I/O in assembly language
    - MIF file format
  - PS5 Emulator implementation
  - PS6 Interfacing with the memory subsystem
    - How to display and set memory contents from inside Quartus – this is limited to the memory accessible in the MIF
  - PS6 Spreadsheet of what happens on each cycle and for each clock edge in sequencer for all instructions

- Preliminary Final Project problem set
- Ed Discussion threads
- Readings

- Anything else

Administrivia (1 of 3)

- **We will *not* be deducting late points from Problem Set grades**

- However, you will still need to complete your final project; so, keep on track
- To complete your final project, it is important for you to be up-to-date with a working assembler, assembly language programs, and emulator, and a finalized ISA documented in a current version of your Principles of Operation manual around now
- Your ALU in VHDL and VHDL code to read then write to the memory subsystem need to be working

Administrivia (2 of 3)

- The Extension School Thanksgiving Break begins tomorrow (Wednesday, November 27th, 2024)

- Have a wonderful Thanksgiving!

Administrivia (3 of 3)

- Our class ski trip will take place on Sunday, January 26th, 2025 at Killington Ski Resort, Vermont
- Please reserve the date!

- We'll furnish more information as the semester progresses

- Killington opened for skiing for pass holders on Thursday, November 14, 2024 and for the general public on Friday, November 15, 2024
- This coming weekend, Killington is hosting the Women's FIS Ski World Cup Giant Slalom and Slalom races (broadcast on Outside TV and NBC) for the eighth year

Coming Attractions

- Section next week: How to use the serial port in the final project
- **Final Project problem set & presentation**
- **Virtual Memory** slides
- **Basic Electronics** slides
- **VLSI Design** slides
- **Future Computer Architecture**
- **Metastability and synchronizers**
- **Dependency analysis**

Problem Sets

- The **VHDL Counter portion of Problem Set 3** and all of **Problem Set 4 (Assembler and Assembly Language Programs)** was due at midnight ET on Sunday, October 27th, 2024

- The **Preliminary Final Project Problem Set (ALU in VHDL)** was due at midnight ET on Sunday, November 3rd, 2024

- **Problem Set 5 (Emulator)** was due at midnight ET on Sunday, November 10th, 2024

- **Problem Set 6 (Sequencer action description, VHDL memory system interaction)** was due at midnight ET on this past Sunday, November 24th, 2024

Final Project Presentations

- Each student will present their final project to the class and visitors during our final class meeting on Tuesday, December 17th, 2024
- Our final class meeting will begin at 6:45 PM ET

- The project presentation meeting is just three weeks from today!

- All final project code, documentation, and presentation material **must** be submitted by **2 PM ET on Friday, December 20th, 2024**

- **We'll discuss final project presentations in detail in class next week**

FPGA/VHDL Implementation

- When you are implementing VHDL for a circuit schematic with MUXes, etc. you don’t need to implement the components from Boolean gates
  - Use all the capabilities that VHDL has to offer
  - Only exceptions: For the **ALU**, design from Boolean gates

FPGA/VHDL Implementation & Memory subsystem requirements

- Remember to **refer back to the presentation in last week's class** of FPGA/VHDL & memory subsystem requirements

New Material for this Week

- Cover new material
  - **Hardware Multiplication** slides
  - **Serial Communication** slides
  - **PDP-8 instruction set** slides
    - **Opcode 6** is for **In-Out Transfer** instructions
  - **PDP-11 instruction set**
    - **PDP-11 Handbook** through the **Central Processor Status Register (PS)** slide
    - **PDP-11 Architecture Handbook**
  - Present **AN5 & AN7** on using the DE2-115 serial port
  - **Floating-Point Number Representation** from the **Numeric Encodings** slides
  - **Virtual Memory** slides

Fourteenth Class Meeting on 12/3/2024

Fourteenth Class Meeting Agenda

- Questions and Comments
- Administrivia
- Review of Current Problem Set Status
- Final Project
- Cover new material
  - **PDP-11 instruction set**
    - **PDP-11 Handbook** beginning after the **Central Processor Status Register (PS)** slide
    - **PDP-11 Architecture Handbook**
  - Present **AN5 & AN7** on using the DE2-115 serial port
  - **Floating-Point Number Representation** from the **Numeric Encodings** slides
  - **Virtual Memory** slides
- Coming attractions
  - **Basic Electronics** slides
  - **VLSI Design** slides
  - **Future Computer Architecture**
  - **Metastability and synchronizers**
  - **Dependency analysis**

Questions and Comments

- Section
- Midterm exam

- Earlier class meetings
  - Your instruction set/assembly language
  - C program for required final demonstration
  - Translating your C program into your assembly language
  - Assembler implementation
    - Dealing with labels and the symbol table
    - Dealing with immediate fields for branches and constants larger than the immediate field size
  - Memory-mapped I/O
  - Altera Memory Initialization File (.mif) format
  - VHDL
  - Clocking Scheme
  - Shifters
  - Comparators
  - Pipelining
  - VHDL
  - Edge-triggered flip-flops
  - Register array
  - Serial communication
  - PDP-8 or introduction to PDP-11 instruction set
  - Hardware Multiplication
  - Caching

- Problem Sets 0 through 6
  - PS3 Counter in VHDL
  - PS4 Assembler, Assembly language programs
    - Dealing with labels and the symbol table
    - Dealing with immediate fields for branches and constants larger than the immediate field size
    - Performing memory-mapped I/O in assembly language
    - MIF file format
  - PS5 Emulator implementation
  - PS6 Interfacing with the memory subsystem
    - How to display and set memory contents from inside Quartus – this is limited to the memory accessible in the MIF
  - PS6 Spreadsheet of what happens on each cycle and for each clock edge in sequencer for all instructions

- Preliminary Final Project problem set
- Ed Discussion threads
- Readings

- Anything else

Administrivia (1 of 2)

- **We will *not* be deducting late points from Problem Set grades**

- However, you will still need to complete your final project; so, keep on track
- To complete your final project, it is important for you to be up-to-date with a working assembler, assembly language programs, and emulator, and a finalized ISA documented in a current version of your Principles of Operation manual around now
- Your ALU in VHDL and VHDL code to read then write to the memory subsystem need to be working

Administrivia (2 of 2)

- Our class ski trip will take place on Sunday, January 26th, 2025 at Killington Ski Resort, Vermont
- Please reserve the date!

- We'll furnish more information as the semester progresses

- Killington opened for skiing for pass holders on Thursday, November 14, 2024 and for the general public on Friday, November 15, 2024
- This coming weekend, Killington is hosting the Women's FIS Ski World Cup Giant Slalom and Slalom races (broadcast on Outside TV and NBC) for the eighth year

Problem Sets

- **All Problem Sets were already due**
- The **VHDL Counter portion of Problem Set 3** and all of **Problem Set 4 (Assembler** and **Assembly Language Programs)** was due at midnight ET on Sunday, October 27th, 2024
- The **Preliminary Final Project Problem Set (ALU in VHDL)** was due at midnight ET on Sunday, November 3rd, 2024
- **Problem Set 5 (Emulator)** was due at midnight ET on Sunday, November 10th, 2024
- **Problem Set 6 (Sequencer action description, VHDL memory system interaction)** was due at midnight ET on this past Sunday, November 24th, 2024

FPGA/VHDL Implementation

- When you are implementing VHDL for a circuit schematic with MUXes, etc. you don’t need to implement the components from Boolean gates
  - Use all the capabilities that VHDL has to offer
  - Only exceptions: For the **ALU**, implement a bit-slice design from Boolean gates

Final Project

- Refer to the Final Project (VHDL processor design) Problem Set

- Produce VHDL code for your entire processor design
  - This includes the data path, the ALU, the memory interface, and the sequencing logic
- The final design should be implemented using the Altera DE2-115 system in conjunction with the furnished memory subsystem

- The culmination of your final project is to run the assembly language code that you wrote for your processor in Problem Set 4, Parts 2 through 9 under your processor on the FPGA
  - In addition, you are welcome to write, run, and demonstrate additional interesting programs running on your hardware
- Demonstrate how you used switches, pushbuttons, LEDs, etc. to assist in tracing/debugging the implementation of your processor
- You should describe and demonstrate the special feature(s) that you implemented

Final Project Class Meeting

- Final class meeting is on Tuesday, December 17th, 2024 starting at 6:45 PM ET and continuing as late as necessary
  - This is **two weeks from today!**
- Sign up for presentation slots **during class next week**
- Students will present their final projects to the class and answer questions
  - Each student will create a ***pre-recorded*** ten minute (maximum) audio & video presentation to be shown to the class that describes and demonstrates their final project
  - An ***additional*** five minutes are reserved for a Q & A session
  - By 4 PM ET on Tuesday, December 17, 2024, all students should send to the course staff the URL for their pre-recorded presentation

Final Project Presentation

- Included in your presentation should be slides that show your:
  - block diagram
  - clocking scheme
  - sequencing logic
  - interesting design aspects of your logic
  - your instruction set
  - assembler
  - emulator
  - special feature(s)
  - a demo of your project running with the memory subsystem
- Show a small number of programs running on your hardware that demonstrate the capabilities of your hardware (including any novel features you have implemented)

Final Project Logistics

- We often have visitors for the final project presentation class meeting
- All students & visitors will participate in the question-and-answer portion

Final Project Code & Document Submission

- By 2 PM ET on Friday, December 20th, 2024, each class member should submit (with git tag term-project) a Final Project report which will include:
  - the slides used during your presentation
  - an overview of the processor
  - an up-to-date copy of the processor block diagram
  - an up-to-date copy of your processor's instruction set (i.e., your Principles of Operation manual)
  - VHDL code for the complete design
  - a current copy of the source code for your assembler
  - a current copy of the source code for your emulator
  - documented sample programs for your processor

Final Project Additional Reminders (1 of 2)

- Please remember to follow all steps in the documentation for the memory subsystem that needs to be compiled with student projects
  - cscie93.sdc file
  - Optimize hold timing
  - Fitter effort
- Remember to follow all information in the CSCI E-93 Application Notes for using the Altera FPGAs
  - AN1: Dealing with DE2-115 Current Strength and Slew Rate
  - AN3: How to load a MIF into Quartus
  - AN4: How to view and alter memory in Quartus
  - AN5: How to run a serial port emulator
  - AN6: How to build a Quartus project that uses the memory subsystem
  - AN7: Notes on Using Serial Port I/O
  - AN8: The Memory Subsystem and Memory Addresses

Final Project Additional Reminders (2 of 2)

- Look in the VHDL slides on how to build a Quartus project
- Remember to **Reserve all unused pins: As input tri-stated**
- Top-level Ports
  - None of the top-level entity port declarations should be changed
  - **Don't declare any** of *your* top-level ports as **inout** or **buffer**
  - All top-level entity ports should be assigned to pins
- Remember to **have an initial reset state** in which nothing happens

Dinner Get-together After Final Presentations

- Meet after the final class meeting for a dinner get-together in Harvard Square
- Significant others are invited to join us

Upcoming Class

- Spring 2025 – CSCI E-95: Compiler Design and Implementation
  - Theory and practice required for the design and implementation of interpreters and compilers for programming languages
  - Topics include lexical analysis, parsing, symbol table generation, type checking, error detection, code generation, optimization, and run-time support
  - Final project is the creation of a compiler for a significant subset of the C Programming Language (ISO C89) that produces code for the MIPS instruction set

New Material for this Week

- Cover new material
  - **PDP-11 instruction set**
    - **PDP-11 Handbook** beginning after the **Central Processor Status Register (PS)** slide
    - **PDP-11 Architecture Handbook**
  - Present **AN5 & AN7** on using the DE2-115 serial port
  - **Floating-Point Number Representation** from the **Numeric Encodings** slides
  - **Virtual Memory** slides

Fifteenth Class Meeting on 12/10/2024

Fifteenth Class Meeting Agenda

- Questions and Comments
- Review of Current Problem Set Status
- Final Project
- Administrivia
- Cover new material
  - **Basic Electronics** slides
  - **VLSI Design** slides
  - **Future Computer Architecture**
  - **Metastability and synchronizers**
  - **Dependency analysis**

Questions and Comments

- Section
- Midterm exam

- Earlier class meetings
  - Your instruction set/assembly language
  - C program for required final demonstration
  - Translating your C program into your assembly language
  - Assembler implementation
    - Dealing with labels and the symbol table
    - Dealing with immediate fields for branches and constants larger than the immediate field size
  - Memory-mapped I/O
  - Altera Memory Initialization File (.mif) format
  - VHDL
  - Clocking Scheme
  - Shifters
  - Comparators
  - Pipelining
  - VHDL
  - Edge-triggered flip-flops
  - Register array
  - Serial communication
  - PDP-8 or introduction to PDP-11 instruction set
  - Hardware Multiplication
  - Caching

- Problem Sets 0 through 6
  - PS3 Counter in VHDL
  - PS4 Assembler, Assembly language programs
    - Dealing with labels and the symbol table
    - Dealing with immediate fields for branches and constants larger than the immediate field size
    - Performing memory-mapped I/O in assembly language
    - MIF file format
  - PS5 Emulator implementation
  - PS6 Interfacing with the memory subsystem
    - How to display and set memory contents from inside Quartus – this is limited to the memory accessible in the MIF
  - PS6 Spreadsheet of what happens on each cycle and for each clock edge in sequencer for all instructions

- Preliminary Final Project problem set
- Ed Discussion threads
- Readings

- Anything else

Problem Sets

- **All Problem Sets were already due**
- The **VHDL Counter portion of Problem Set 3** and all of **Problem Set 4 (Assembler** and **Assembly Language Programs)** was due at midnight ET on Sunday, October 27th, 2024
- The **Preliminary Final Project Problem Set (ALU in VHDL)** was due at midnight ET on Sunday, November 3rd, 2024
- **Problem Set 5 (Emulator)** was due at midnight ET on Sunday, November 10th, 2024
- **Problem Set 6 (Sequencer action description, VHDL memory system interaction)** was due at midnight ET on this past Sunday, November 24th, 2024

Late Point Deductions

- **We will *not* be deducting late points from Problem Set grades**

- To complete your final project, you need to already be up-to-date with a working assembler, assembly language programs, and emulator, a finalized ISA documented in a current version of your Principles of Operation manual, the PS6 memory access code, and the PS6 sequencer spreadsheet

FPGA/VHDL Implementation

- When you are implementing VHDL for a circuit schematic with MUXes, etc. you don’t need to implement the components from Boolean gates
  - Use all the capabilities that VHDL has to offer
  - Only exceptions: For the **ALU**, design from Boolean gates

Final Project

- Refer to the Final Project (VHDL processor design) Problem Set
- Refer to all relevant material covered in class and section meetings, slide decks covered in class, sample VHDL code, class Application Notes, class Laboratory Documents and Programs – all of which are available through the class website

- Produce VHDL code for your entire processor design
  - This includes the data path, the ALU, the memory interface, and the sequencing logic
- The final design should be implemented using the Altera DE2-115 system in conjunction with the furnished memory subsystem

- The culmination of your final project is to run the assembly language code that you wrote for your processor in Problem Set 4, Parts 2 through 9 under your processor on the FPGA
  - In addition, you are welcome to write, run, and demonstrate additional interesting programs running on your hardware
- Demonstrate how you used switches, pushbuttons, LEDs, etc. to assist in tracing/debugging the implementation of your processor
- You should describe and demonstrate the special feature(s) that you implemented

Final Project Class Meeting

- Final class meeting is on Tuesday, December 17th, 2024 starting at 6:45 PM ET during section and class time
  - This is **next week!**
- Sign up for a presentation slot *momentarily*
- Students will present their final projects to the class and answer questions
  - Each student will create a ***pre-recorded*** **ten minute (maximum) audio & video presentation** to be shown to the class that describes and demonstrates their final project
  - By **4 PM ET on Tuesday, December 17th, 2024**, we need to have received the URL of your **ten minute** video presentation
  - An ***additional*** five minutes are reserved for a Q & A session
- We often have visitors present for the presentations
- Your term project grade includes the video presentation

Final Project Presentation (1 of 2)

- Included in your presentation should be slides that show your:
  - block diagram
  - clocking scheme
  - sequencing logic
  - interesting design aspects of your logic
  - your instruction set
  - assembler
  - emulator
  - special feature(s)
  - a demo of your project running with the memory subsystem
- Show a small number of programs running on your hardware that demonstrate the capabilities of your hardware (including any novel features you have implemented)

Final Project Presentation (2 of 2)

- Even if your processor is not fully-functional, make a presentation that includes the work that you have completed
  - Principles of Operation
  - Block Diagram
  - ISA (Instruction Set Architecture)
  - Assembler
  - Emulator
  - etc.

Final Project Logistics

- We often have visitors for the final project presentation class meeting
- All students & visitors will participate in the question-and-answer portion

Final Project Code & Document Submission

- **By 2 PM ET on Friday, December 20th, 2024**, each class member should submit (with git tag term-project) a Final Project report which will include:
  - the slides used during your presentation
  - an overview of the processor
  - an up-to-date copy of the processor block diagram
  - an up-to-date copy of your processor's instruction set (i.e., your Principles of Operation manual)
  - VHDL code for the complete design
  - a current copy of the source code for your assembler
  - a current copy of the source code for your emulator
  - documented sample programs for your processor

Final Project Additional Reminders (1 of 2)

- Please remember to follow all steps in the documentation for the memory subsystem that needs to be compiled with student projects
  - cscie93.sdc file
  - Optimize hold timing
  - Fitter effort
- Remember to follow all information in the CSCI E-93 Application Notes for using the Altera FPGAs
  - AN1: Dealing with DE2-115 Current Strength and Slew Rate
  - AN3: How to load a MIF into Quartus
  - AN4: How to view and alter memory in Quartus
  - AN5: How to run a serial port emulator
  - AN6: How to build a Quartus project that uses the memory subsystem
  - AN7: Notes on Using Serial Port I/O
  - AN8: The Memory Subsystem and Memory Addresses

Final Project Additional Reminders (2 of 2)

- Look in the VHDL slides on how to build a Quartus project
- Remember to **Reserve all unused pins: As input tri-stated**
- Top-level Ports
  - None of the top-level entity port declarations should be changed
  - **Don't declare any** of *your* top-level ports as **inout** or **buffer**
  - All top-level entity ports should be assigned to pins
- Remember to **have an initial reset state** in which nothing happens

Final Project Presentation Schedule on **Tuesday, December 17, 2024** (All Eastern Time)

- Slot 1, 6:45 PM-7:00 PM ET: Mikey
- Slot 2, 7:00 PM-7:15 PM ET: Nathan
- Slot 3, 7:15 PM-7:30 PM ET: Patrick
- Slot 4, 7:30 PM-7:45 PM ET: Ben
- Slot 5, 7:45 PM-8:00 PM ET: (Leonid, tentative)
- Slot 6, 8:00 PM-8:15 PM ET: Jonathan
- Slot 7, 8:15 PM-8:30 PM ET: Julian
- Slot 8, 8:30 PM-8:45 PM ET: Aliya
- Slot 9, 8:45 PM-9:00 PM ET:
- Slot 10, 9:00 PM-9:15 PM ET:
- Slot 11, 9:15 PM-9:30 PM ET: (Frederic, tentative for the last slot)

Course Evaluation

- Please fill out the course evaluation form that will soon be available to you

- Course Evaluation is available on-line from December 12, 2024 to January 2, 2025

Dinner Get-together After Final Presentations

- Meet after the final class meeting for a dinner get-together in Harvard Square
- Significant others are invited to join us

Upcoming Class

- Spring 2025 – CSCI E-95: Compiler Design and Implementation
  - Theory and practice required for the design and implementation of interpreters and compilers for programming languages
  - Topics include lexical analysis, parsing, symbol table generation, type checking, error detection, code generation, optimization, and run-time support
  - Final project is the creation of a compiler for a significant subset of the C Programming Language (ISO C89) that produces code for the MIPS instruction set

Ski Trip Between Semesters

- Our class ski trip will take place on Sunday, January 26th, 2025 at Killington Ski Resort, Vermont
- Please reserve the date!

- We'll furnish more information via e-mail after the semester is over

New Material for this Week (1 of 3)

- Cover new material
  - **Basic Electronics** slides
- Show & Tell
  - **Altera DE2-115 Printed Circuit Motherboard with Hermetically Sealed IC Packages**
  - **Wafer**
  - **Etched Wafer**
  - **Chip**
  - **Chip in Open Ceramic Header**
- Cover new material
  - **VLSI Design** slides

New Material for this Week (2 of 3)

- **Processor Photomicrographs** (See https://en.wikipedia.org/wiki/Transistor_count)
  - **Intel 4004 (4-bit CPU, 1971, 2,300 transistors)**
  - **Intel 8008 (8-bit CPU, 1972, 3,500 transistors)**
  - **Intel 8080 (8-bit CPU, 1974, 6,000 transistors)**
  - **Intel 8085 (8-bit CPU, 1976, 6,500 transistors)**
  - **Intel 8088 (16-bit CPU, 1979, 29,000 transistors)**
  - **Intel 80286 (16-bit CPU, 1982, 134,000 transistors)**
  - **Intel 80386DX, i387, 486, 486DX2**
  - **Intel i960 Cobra (32-bit CPU, 1984)**
  - **Intel Prescott Pentium 4 (64-bit, 2004, 90nm, 125,000,000 transistors, 16K 8-way associative L1 cache, 1MB L2 cache, 103W)**
  - **Intel Haswell-E Core i7-5960X (64-bit, 2014, 22nm, 2,600,000,000 transistors, 8 cores, 16 processor threads, 8x 256KB L2 cache, 20MB L3 cache, 140W, 40 PCIe 3.0 lanes)**
  - **Intel Coffee Lake R i9-9900K (64-bit, 2018, 14nm, 3,000,000,000 transistors, 8 cores, 1,536 µOPs 8-way associative L0 µOP cache, 32 KB 8-way associative L1I Cache, 32 KiB, 8-way associative L1D Cache, Unified 256 KB, 4-way associative L2 Cache, Up to 2 MB/core (shared across all cores) up to 16-way associative L3 Cache/LLC, L1 TLB for instruction (ITLB) and another one for data (DTLB), Additionally there is a unified L2 TLB (STLB), 95W)**

New Material for this Week (3 of 3)

- Cover new material
  - **Future Computer Architecture**
  - **Metastability and synchronizers**
  - **Dependency analysis**

Sixteenth Class Meeting on 12/17/2024

Sixteenth Class Meeting Agenda

- Administrivia
- Final Project Presentations

Course Evaluation

- Please fill out the course evaluation form that is currently available to you

- Course Evaluation is available on-line from December 12, 2024 to January 2, 2025

Dinner Get-together After Final Presentations

- Meet after class tonight for a dinner get-together in Harvard Square
- Significant others are invited to join us

Upcoming Class

- Spring 2025 – CSCI E-95: Compiler Design and Implementation
  - Theory and practice required for the design and implementation of interpreters and compilers for programming languages
  - Topics include lexical analysis, parsing, symbol table generation, type checking, error detection, code generation, optimization, and run-time support
  - Final project is the creation of a compiler for a significant subset of the C Programming Language (ISO C89) that produces code for the MIPS instruction set

Ski Trip Between Semesters

- Our class ski trip will take place on Sunday, January 26th, 2025 at Killington Ski Resort, Vermont
- Please reserve the date!

- We'll furnish more information via e-mail after the semester is over

Final Project Presentations