EECS 583 – Class 15
Exam Review

University of Michigan

October 27, 2025

Announcements

- Project proposal deadline – Tonight (Monday) midnight
  - Submit paragraph + reference on your project topic (Email to Naveen, Rishika, Scott, and cc all your group members)
- Research paper presentations
  - Each group sign up for 15 min (presentation) + 2 min (Q&A) slot on the EECS 583 calendar
  - Mon Nov 3 – Wed Dec 3: presentations during class (4 slots per class)
- Midterm Exam
  - Wednesday, Oct 29, Hybrid format
  - In person – Send email to Naveen if you want to take the exam in class
    - 10:35am-12:05pm – 1500 EECS
    - Questions answered in the hallway
  - Virtual
    - 10:35am-12:05pm + 15 mins extra time for logistics (printing, scanning, etc.)
    - Questions about exam can be posted on piazza and will be answered ASAP
  - Covers through register allocation (last lecture)

Research Paper Presentation Logistics

- Monday Nov 3 – Wednesday Dec 3
  - Signup for slot on Google calendar (just like project proposals)
  - Sign up for earliest slot available on the day you want to present → no gaps
    - Plan on attending the entire lecture on the day you present
  - Not all days will be full (max of 4 slots per lecture)
- Each group: 15 min slot + 2 mins Q&A
  - You will be cut off if you go long!
  - Tag-team presentation – Divide up as you like but everyone must talk
  - Max of 20 slides (for the group), animations not included in count
  - Record your paper title on the group sign up sheet (so we can check for conflicts!)
  - Submit paper (pdf) and slides (pptx or pdf) night before (by 9pm!)
    - Call your files groupX_paper.pdf, groupX_slides.pdf/pptx
    - Email to Scott

Research Paper Presentation Format

- Make your own slides!!!
  - Don’t just lift figures from the pdf (graphs/tables ok to lift)
  - Don’t have too many all text slides
  - No long sentences on slides, don’t just read the slides
- Points to discuss
  - Intro/Motivation – area + problem + why is it important to solve this problem
  - How the technique works, examples are super helpful
  - Some results, but don’t show 10 graphs
  - Group’s commentary (last slide or 2 of your presentation)
    - What is best about the paper? Why is the idea so awesome? Don’t focus on results
    - What are limitations/weaknesses of the approach (be critical!)

Research Paper Presentation – Audience Members

- Research presentations != skip class
  - You should attend or watch the Zoom video
- Grading + give comments to your peers
  - Class + Naveen, Rishika, & I will evaluate each group's presentation and provide feedback
  - Each person will submit evaluation sheet for the day's presentations
    - Online Google form
    - 3 days (72 hrs) to submit
  - Naveen/Rishika will anonymize comments and email to each group
  - Be critical, but constructive with your criticisms
    - What was good about the talk, what could be improved.
    - Don't try to give separate comments for each group member, just evaluate the entire team

Exam Review

Virtual Exam Logistics (see piazza for more details)

- Wednesday, Oct 29
- 10:35-12:05 + 15 minutes for logistics
- Gradescope to distribute/collect exams, accessible via canvas
- Steps
  - Download, take exam, scan & submit
    - Print out and write on exam sheets – then scan your sheets and submit
    - Use electronic method (ie tablet) to create electronic answers
  - Exam itself should take ~ 75 mins (90 mins provided)
  - Some slack time to deal with difficulties, but email course staff if you run into problems
- Use piazza to ask private questions and get answers during the exam
  - We will answer ASAP.

In-person Exam Logistics

- Wednesday Oct 29 – 1500 EECS
- 10:35-12:05
- Printed exams available in classroom
- Logistics
  - Take exam in room
  - Course staff will be outside lecture room to answer questions
- Bring whatever you like (open book exam)
  - Tablet/laptop
  - Printed materials (old exams, lecture problems, etc.)
  - Books, etc.

What to Expect

- Exam format
  - Open notes, open internet
  - Apply techniques we discussed in class
  - Reason about solving compiler problems – how/why things are done
  - A couple of thinking problems
  - No LLVM code
- Honor code and cheating
  - Must sign honor code acknowledging that you have neither given no received aid on the exam – this includes ChatGPT or other LLM
  - Please do not share answers or talk to other students during the exam
  - Graduate class, so we don’t expect cheating to be an issue
    - But we will investigate any anomalies that arise

Studying

- 10 exams + answer keys (F12, F13, F18, F19, F20, F21, F22, W23, F23, W24) are posted on the course website
  - Note – Past exams may not accurately predict future exams!!
  - Fomat will be similar
  - Work out the problems without looking at the answers!
  - **Exams vary in terms of time/length**
- Preparing yourself
  - Yes, you should study even though its open notes
    - Lots of material that you have likely forgotten from early this semester
    - Refresh your memories, especially the old topics
    - No memorization required, but you need to be familiar with the material to finish the exam
  - Go through lecture notes, especially the examples!
  - If you are confused on a topic, go through the reading
  - Go through the practice exams (Don't look at the answer) as the final step

Exam Topics

- Control flow analysis
  - Control flow graphs, Dom/pdom, Loop detection
  - Trace selection, superblocks
- Predicated execution
  - Control dependence analysis, if-conversion
- Dataflow analysis
  - Liveness, reaching defs, DU/UD chains, available defs/exprs
  - Static single assignment – Make sure you understand SSA!
- Optimizations
  - Classical: Dead code elim, constant/copy prop, CSE, LICM, induction variable strength reduction
  - ILP optimizations - unrolling, tree height reduction, induction/accumulator expansion – Just understand the concepts
  - Speculative optimization – like HW2

Exam Topics - Continued

- Acyclic scheduling
  - Dependence graphs, Estart/Lstart/Slack, list scheduling
  - Code motion across branches, speculation, exceptions
- Software pipelining
  - DSA form, ResMII, RecMII, modulo scheduling
  - Make sure you can modulo schedule a loop!
  - Execution control with LC, ESC
- Register allocation
  - Interference graph, graph coloring

Some Sample Problems

Part I: Short Questions

- Fast questions – a couple of minutes each
  - Don’t waste too much time on any single question
  - Come back later to questions you don’t know the answer
- Basic facts/trends, simple problems
- Most should be obvious, but read carefully

Question 2 - Winter 2023

- In liveness analysis, how would the analysis results be changed if the meet function was modified from using union of live variables (IN sets) of the successor blocks to intersection over the same sets of variables? Briefly explain your answer

Question 2 – Fall 2022

- What is the main purpose of a compiler identifying and co-locating hot blocks of code together as done with trace selection?

Question 4 – Winter 2024

- Using Homework 2 as inspiration, would frequent-path constant propagation provide any advantages over traditional constant propagation? Briefly explain with a small code example

Question 3 – Fall 2018

- When a compiler scheduler wants to speculate an instruction, name one issue that it must consider to preserve correctness of the resulting code.

Question 2 – Fall 2020

- It is often possible to improve the performance of a loop limited by RecMII by adding resources to the processor. Is the preceding statement True or False? Justify your answer

Question 3 – Fall 2021

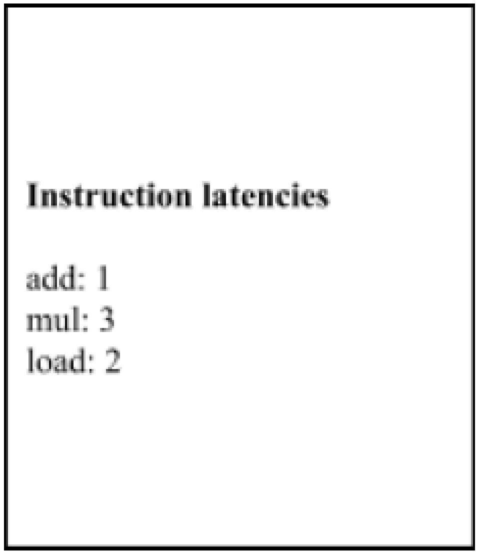
- Is it possible to unroll a loop with a statically (compile-time) unknown number of iterations? Yes/No and briefly explain.

Part II: Medium/Long Questions

- Longer questions
  - Problems that must be worked out: 5-10 mins each
  - Some questions like lecture examples
  - But, some have a little twist
- Practicing problems ahead of time will make you more comfortable and faster
  - So, practicing is strongly recommended
  - Go through the old exams!

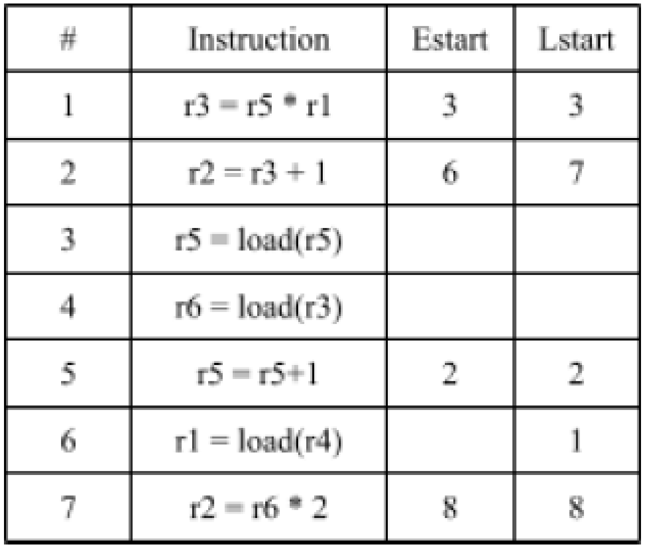
Question 9 – Fall 2022

Due to a corrupt disk, the original order of the instructions was lost and the instructions got randomly ordered. The student reassigns the number of each instruction and knows the corresponding partial Estart and Lstart values (see Table below). It is also known that Instruction 7 (r2 = r6*2) is the last instruction of the BB and has the largest Estart and Lstart values. Determine the original order of the instructions using the partial Estart/Lstart values and complete the missing Estart/Lstart values in the table below for the original ordering. Remember, the instruction numbers do not represent the original order.



Instruction latencies

add: 1
mul: 3
load: 2



#	Instruction	Estart	Lstart
1	r3 = r5 * r1	3	3
2	r2 = r3 + 1	6	7
3	r5 = load(r5)		
4	r6 = load(r3)		
5	r5 = r5+1	2	2
6	r1 = load(r4)		1
7	r2 = r6 * 2	8	8

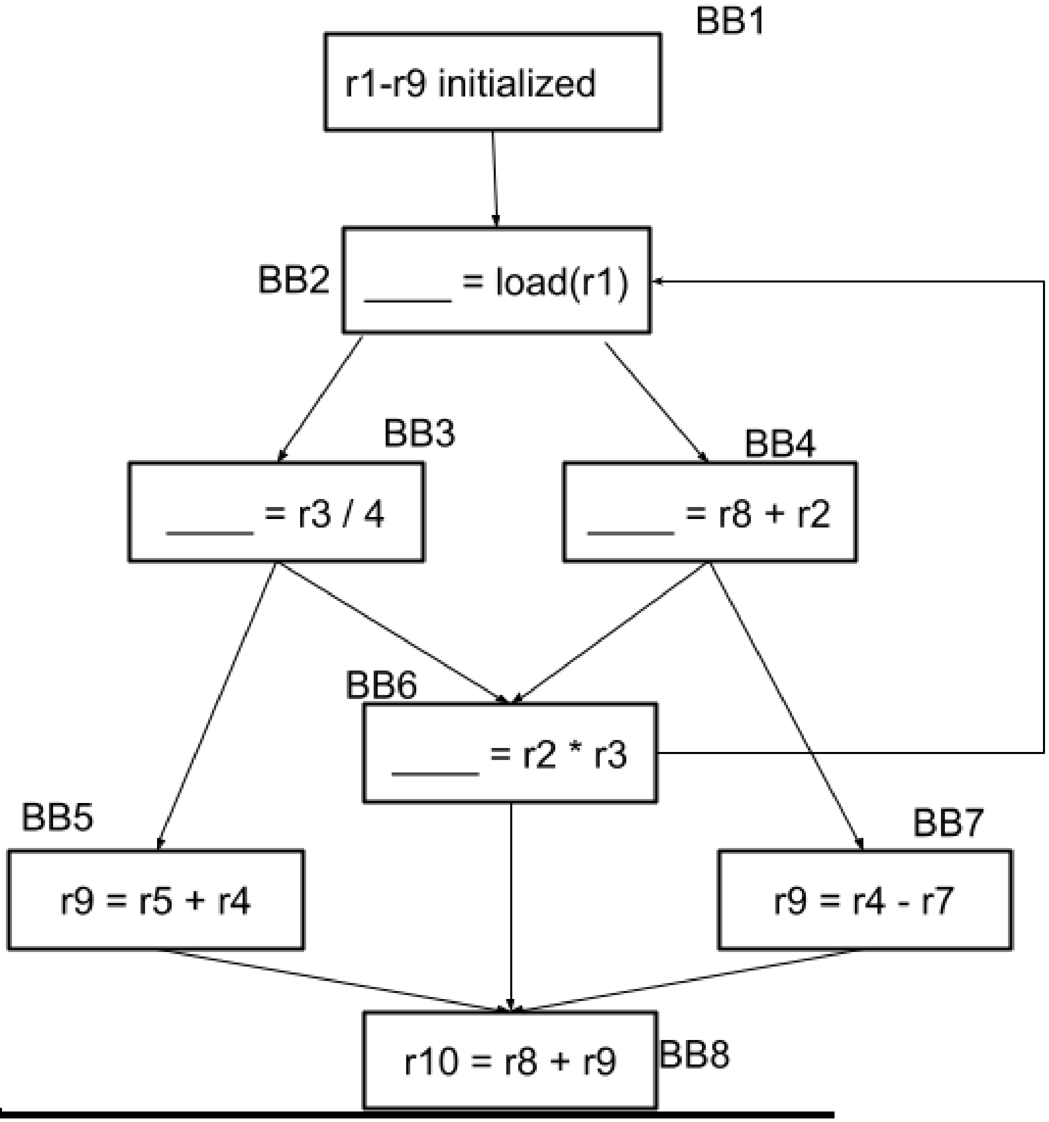
Question 6 – Fall 2019

Draw the control flow graph (CFG) and determine
the *minimum* number of predicates required to if-convert the following code.

```
do {
   if (a>0 && b>0){
      if (c>0)
         x+=1;
      else
         x+=2;
      z=x/3;
   }else{
      y+=1;
   }
} while(z<100)
```

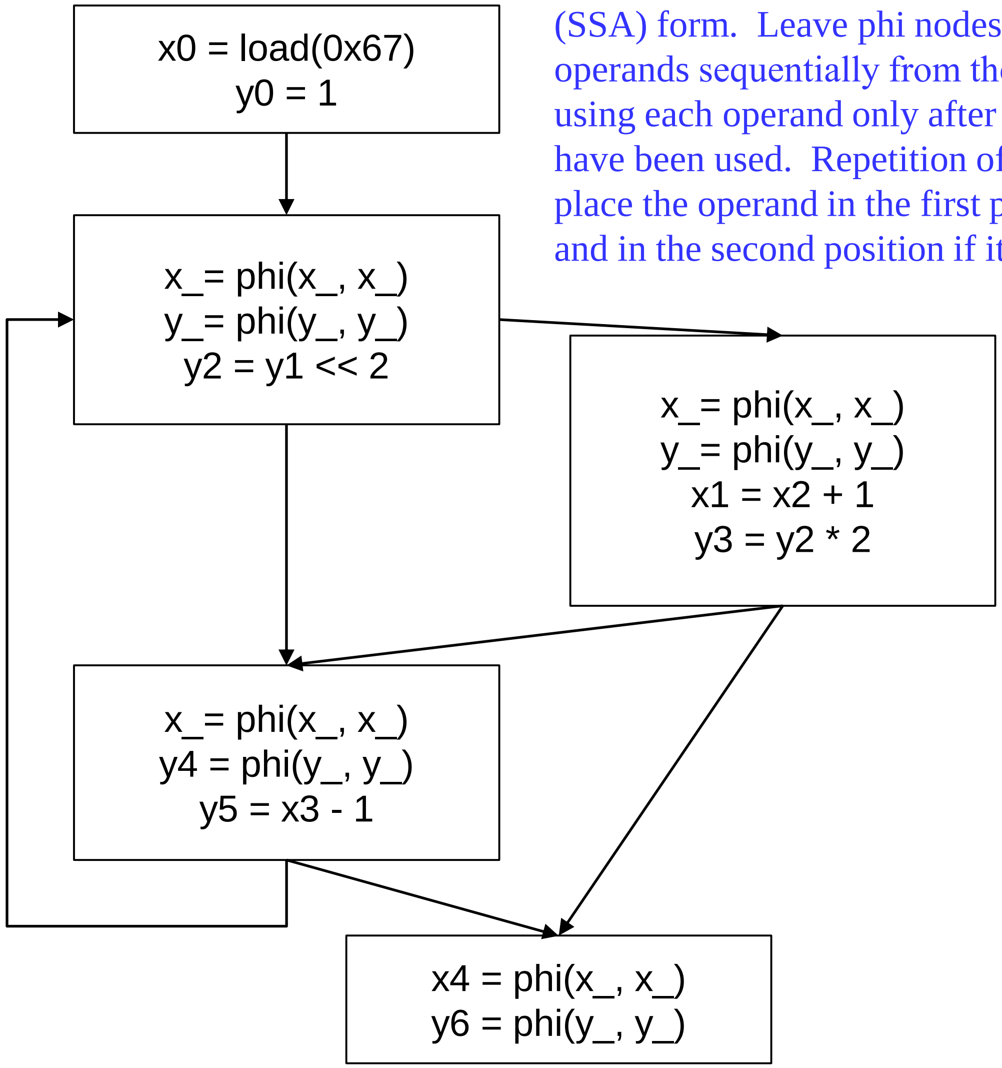
Question 7 – Fall 2023

Fill in the blanks using r4, r5, r7, and r8, so that a maximum number of instructions from BB2, BB3, BB4, and BB6 become eligible for hoisting via LICM



Question 9 – Winter 2024

Fill in the blanks below to put the code into the static single assignment (SSA) form. Leave phi nodes blank when they are unnecessary. Choose operands sequentially from the lists x0, x1, … xn, and y0, y1, … ym, using each operand only after all its preceding operands in the sequence have been used. Repetition of registers used is permitted. For phi nodes, place the operand in the first position if it originates from the left edge, and in the second position if it originatesfrom the right edge.



x0 = load(0x67)
y0 = 1

x_= phi(x_, x_)
y_= phi(y_, y_)
y2 = y1 << 2

x_= phi(x_, x_)
y_= phi(y_, y_)
x1 = x2 + 1
y3 = y2 * 2

x_= phi(x_, x_)
y4 = phi(y_, y_)
y5 = x3 - 1

x4 = phi(x_, x_)
y6 = phi(y_, y_)