EECS 583 – Class 15
Exam Review

University of Michigan

March 13, 2023

Announcements

- Project proposal deadine – Tonight (Monday) midnight
  - Submit paragraph + reference on your project topic (Email to Aditya and Scott)
- Research paper presentations
  - Each group sign up for 15+5 min slot on the EECS 583 calendar
  - Mon Mar 20 – Wed Apr 12: presentations during class
- Midterm Exam
  - Wednesday, Mar 15, Hybrid format
  - In person – Send email to Aditya/I if you want to take the exam in class
    - 10:30am-11:50am
    - Questions answered in the hallway
  - Virtual
    - 10:30am-11:50am + 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 Mar 20 – Wednesday Apr 12
  - 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 + 5 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
  - Submit paper (pdf) and slides (pptx or pdf) night before (by 9pm)
    - Email to Aditya and 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 + Aditya & 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
  - Aditya 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 Mar 15
- 10:30-11:50 + 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
    - Just write answers on paper
    - Use electronic method (ie tablet) to create electronic answers
  - Exam itself should take ~ 60-70 mins
  - Some slack time to deal with difficulties, but email course staff if you run into problems
- Use piazza to ask questions and get answers during the exam
  - We will answer ASAP. Be sure to read others questions before posting your own.

In-person Exam Logistics

- Wednesday Mar 15 – G906 Cooley
- 10:30-11:50
- Printed exams available in classroom
- Steps – Normal pre-COVID exam
  - Exam itself should take 60-70 mins
  - Course staff will be outside lecture room to answer questions
- Bring whatever you like
  - 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
  - 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

- 7 exams (F12-F13, F18, F19, F20, F21, F22) 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!
  - **We will have less time than previous exams, so ours will be shorter than prior exams**
- 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

Work Out 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
- Most should be obvious, but some a little thought

Question 1 – Fall 2021

- What is the main difference between reaching definitions and available definitions?

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 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

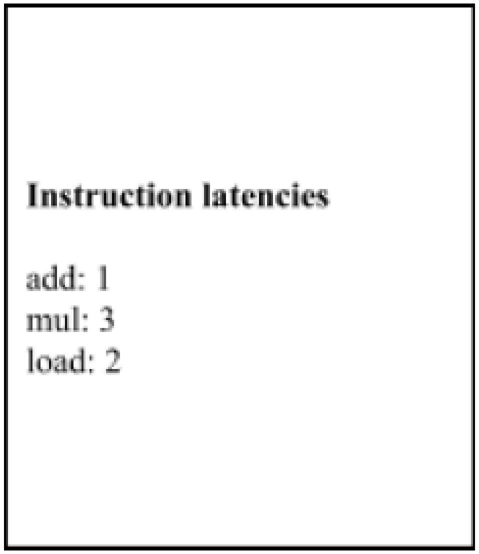
- Is it possible to unroll a loop with a statically (compile-time) unknown number of iterations? Yes/No and briefly explain.

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

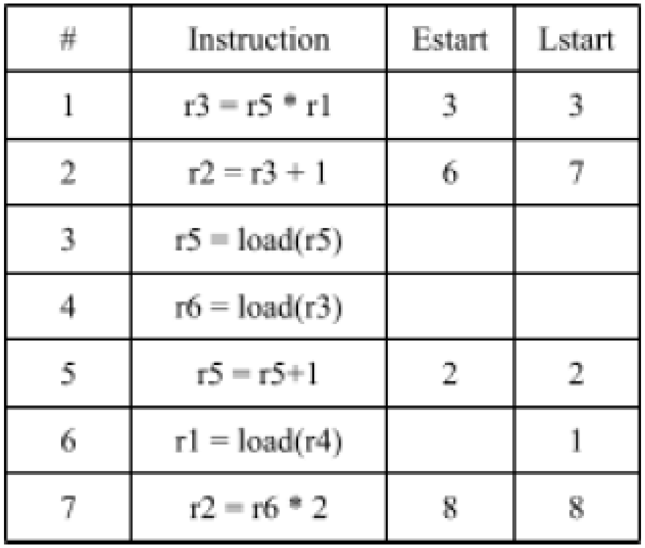
Question 9 – Fall 2022

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

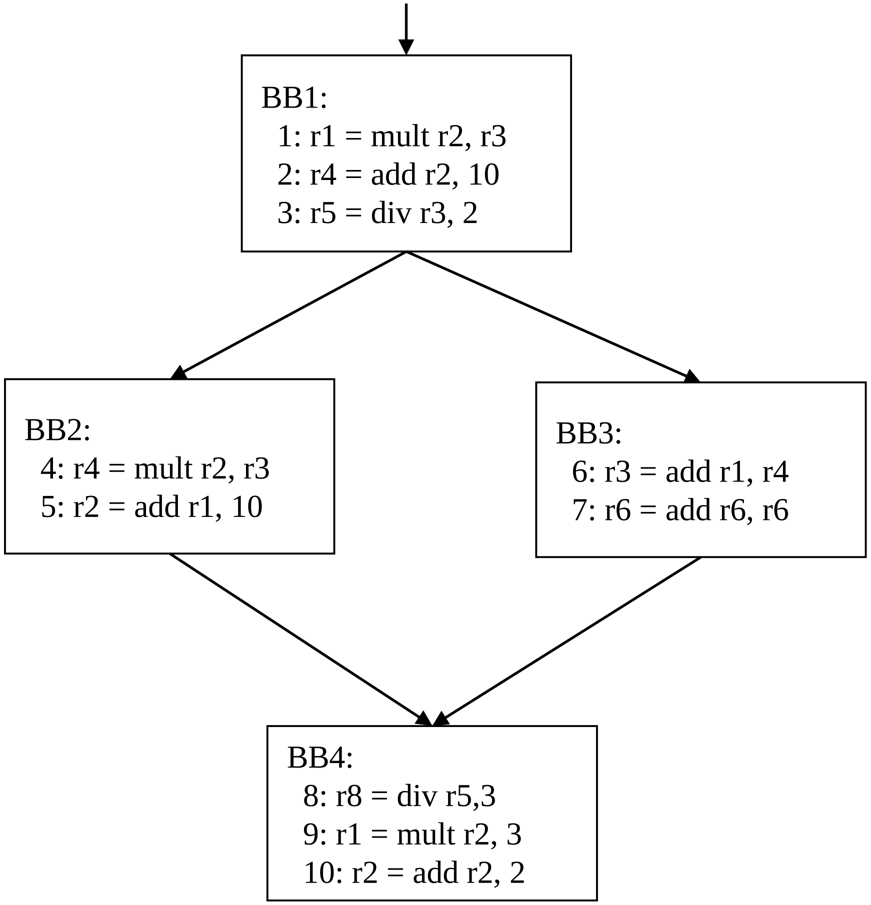
Question 6 – Fall 2019

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 2020

Compute the Available Expression GEN/KILL/IN/OUT sets at BB4.
Assume r2, r3, r6 are defined before entering BB1



```
BB1:
  1: r1 = mult r2, r3
  2: r4 = add r2, 10
  3: r5 = div r3, 2
```

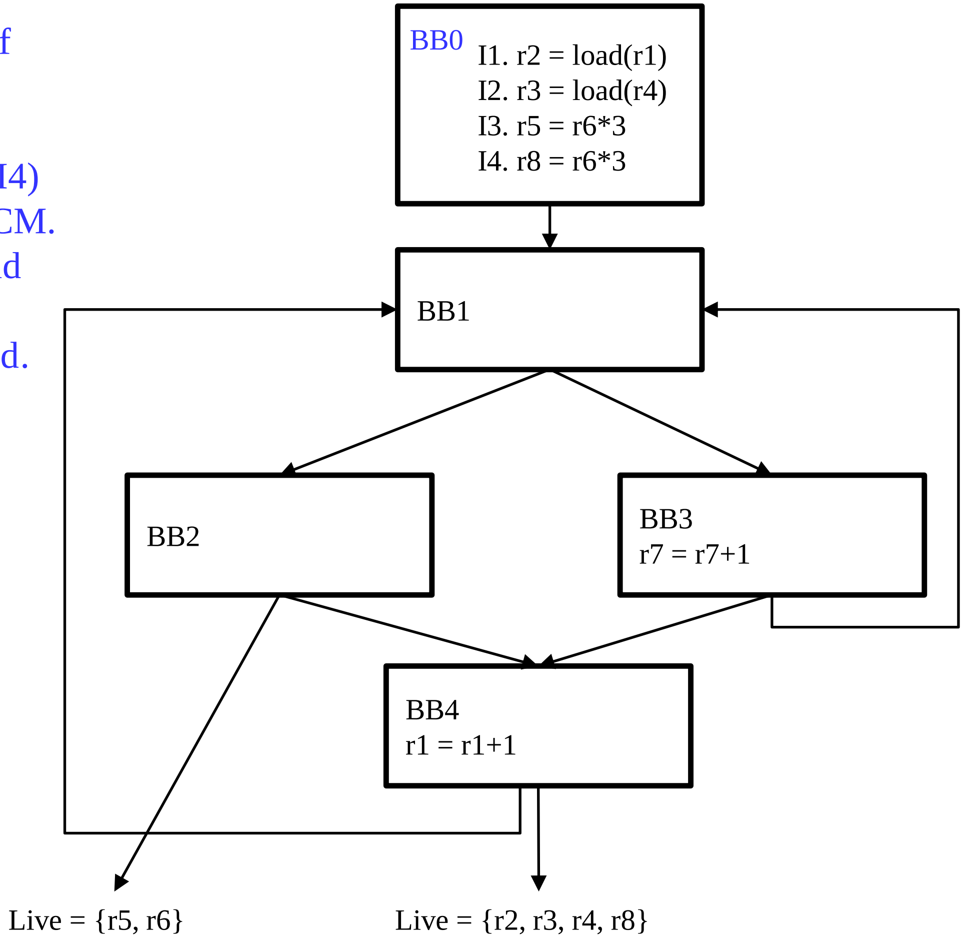
```
BB2:
  4: r4 = mult r2, r3
  5: r2 = add r1, 10
```

```
BB3:
  6: r3 = add r1, r4
  7: r6 = add r6, r6
```

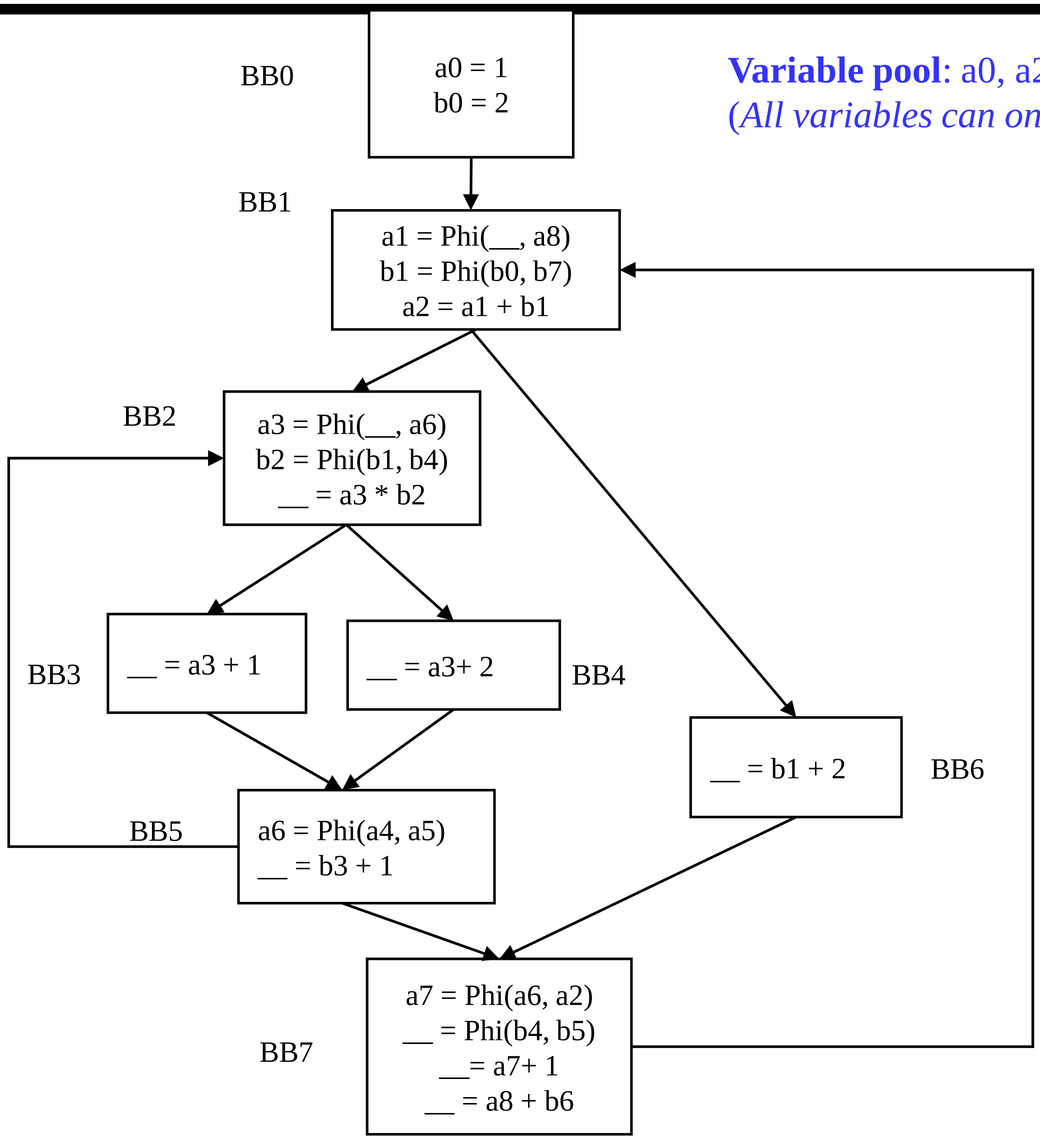
```
BB4:
  8: r8 = div r5,3
  9: r1 = mult r2, 3
  10: r2 = add r2, 2
```

Question 8 – Fall 2019

You are trying to reverse engineer some optimized assembly code to determine the original locations of instructions before optimization. In the following loop consisting of 4 basic blocks (BB1-BB4), the preheader (BB0) contains 4 instructions (I1, I2, I3, I4) that were possibly removed from the loop using LICM. For each instruction, determine whether LICM could have been legally applied and if so, which basic block(s) the instruction could have originally resided.



Question 9 – Fall 2020



Variable pool: a0, a2, a4, a5, a8; b3, b4, b5, b6, b7
(*All variables can only be used once*)

```
BB0
a0 = 1
b0 = 2

BB1
a1 = Phi(__, a8)
b1 = Phi(b0, b7)
a2 = a1 + b1

BB2
a3 = Phi(__, a6)
b2 = Phi(b1, b4)
__ = a3 * b2

BB3
__ = a3 + 1

BB4
__ = a3+ 2

BB5
a6 = Phi(a4, a5)
__ = b3 + 1

BB6
__ = b1 + 2

BB7
a7 = Phi(a6, a2)
__ = Phi(b4, b5)
__= a7+ 1
__ = a8 + b6
```

Blank Slide

Blank Slide