Final

ⓘ This is a preview of the published version of the quiz

Started: Dec 20 at 9:48am

Quiz Instructions

Regulations: https://www.seas.upenn.edu/~ese532/fall2020/final_details.pdf

---

Question 1 — 1 pts

I certify that I have complied with the University of Pennsylvania's Code of Academic Integrity and the exam regulations [https://www.seas.upenn.edu/~ese532/fall2020/final_details.pdf (https://www.seas.upenn.edu/~ese532/fall2020/midterm_details.pdf)](https://www.seas.upenn.edu/~ese532/fall2020/final_details.pdf) in completing this exam.

- ◯ True
- ◯ False

Consider the following application in answering the questions on this exam:

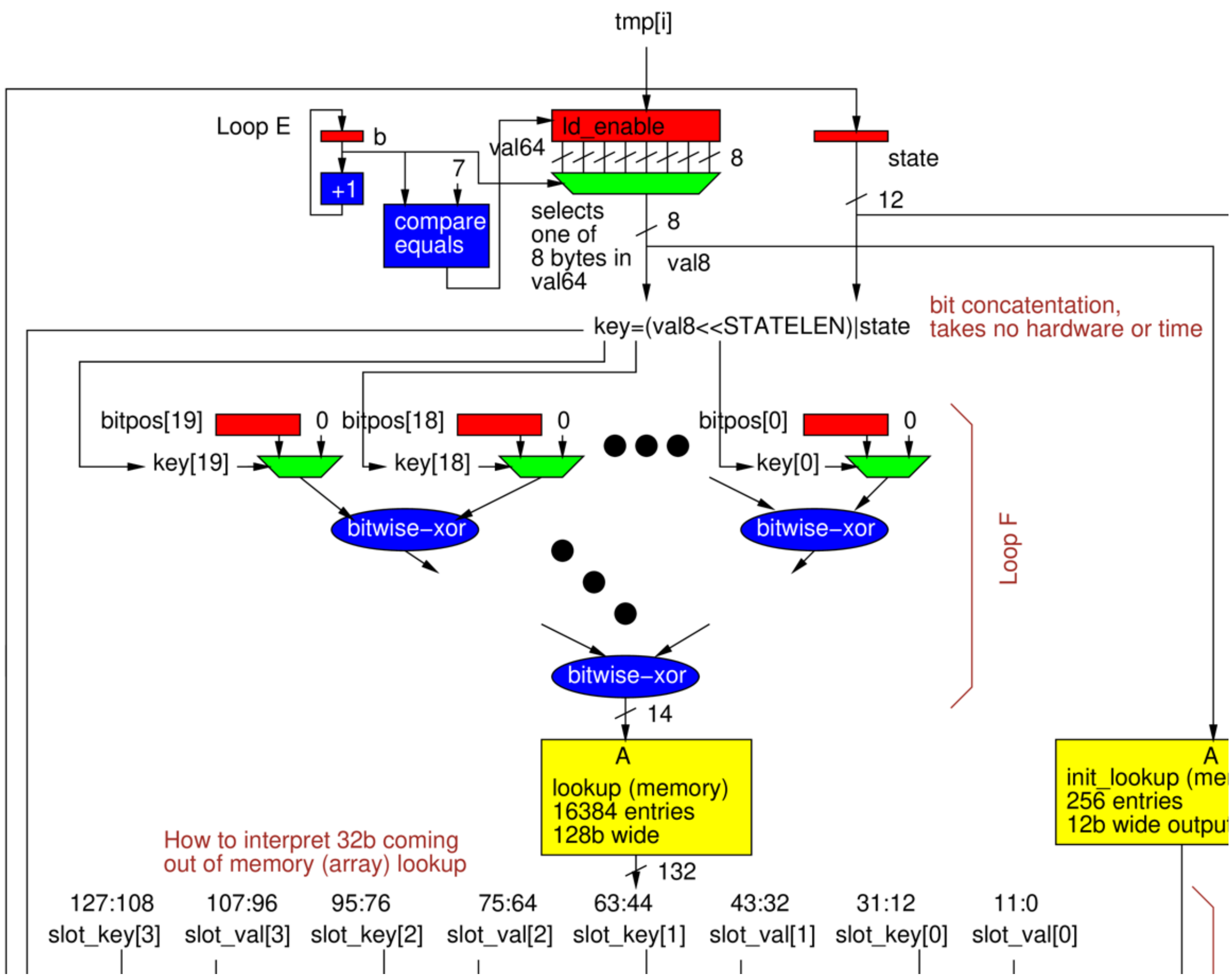
```
#define NFRAMES 256
#define MAX_RESULTS 8*NFRAMES
#define frame_type ap_int<240>
#define lookup_result_type ap_int<128>
#define STATELEN 12
#define KEYLEN (STATELEN+8)
#define KEY_MASK 0x0FFFFF
```

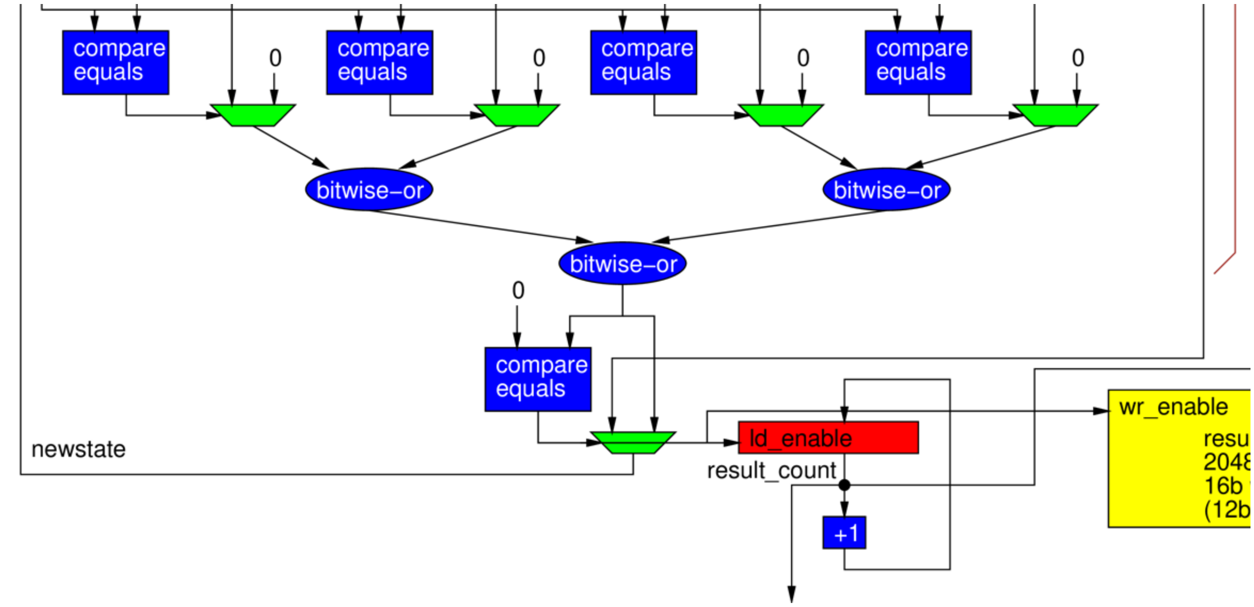
```
#define KEY_MASK 0x0FFFFF
#define VAL_MASK 0x0FFF
#define NUM_SLOTS 16384
#define BUCKET_MASK 0x0FFFFFFF
#define slot_type uint32_t
#define BUCKET_CAPACITY 4
#include<stdint.h>
extern lookup_result_type lookup[NUM_SLOTS];
extern uint16_t init_lookup[256];
void extract_compress(frame_type frames[NFRAMES],
                      uint8_t bitlocs[64],
                      uint16_t bitpos[KEYLEN],
                      uint16_t results[MAX_RESULTS],
                      int *num_results)
{
 uint64_t tmp[NFRAMES];

 for (int i=0;i<NFRAMES;i++) { // Loop A
   uint64_t result=0;
   int finalpos=1;
   frame_type val=frames[i];
   for (int j=0;j<64;j++) { // Loop B
     uint8_t bitloc=bitlocs[j];
     for (int k=128;k>0;k=k/2) { // Loop C
         if ((bitloc&0x01)==1)
             val=val/k;
         bitloc=bitloc/2;
      }
     if ((val&0x01)==1)
         result|=finalpos;
     finalpos=finalpos*2;
    }
   tmp[i]=result;
 }
 int result_count=0;
 int state=0;
 for (int i=0;i<NFRAMES;i++) { // Loop D
   uint64_t val64=tmp[i]; // val64 is input to pipeline pix
   for (int b=0;b<8;b++)  // Loop E (also shown in pipeline pix)
```

```
    {
      <see pipeline in next box; complete code for question 4>
    }
  }
  *num_results=result_count;
  return;
}
```

Here is a pipeline that implements loop E and the code (and loops) inside it.

Memories **bitpos**, **lookup**, **init_lookup**, and **results** are arrays defined in the code above. Registers **val64**, **b**, **state**, and **result_count** are variables defined in the code above.





Question 2

5 pts

Assuming:

- up to 6-input xors or ors can be packed into a single 1ns operation
- a comparison followed by a mux can be performed in a single 1ns operation
  - mux alone can also be performed in a 0.2ns
- each memory operation can be performed in a single 2ns operation
- an add or increment can occur in a single 1ns operation
- bitpos registers are preloaded

What is the cycle-bound II for loop E in ns?

Question 3

5 pts

Explain your II answer to previous question.

HTML Editor

12pt

Paragraph

0 words

Question 4

10 pts

Provide code for the body of Loop E based on the pipeline show.

Recreate loops as flagged in the diagram.

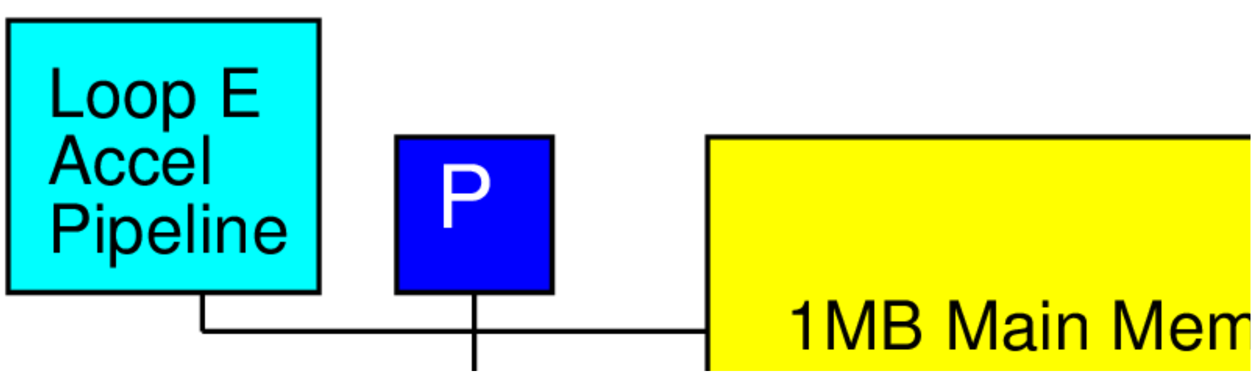
HTML Editor

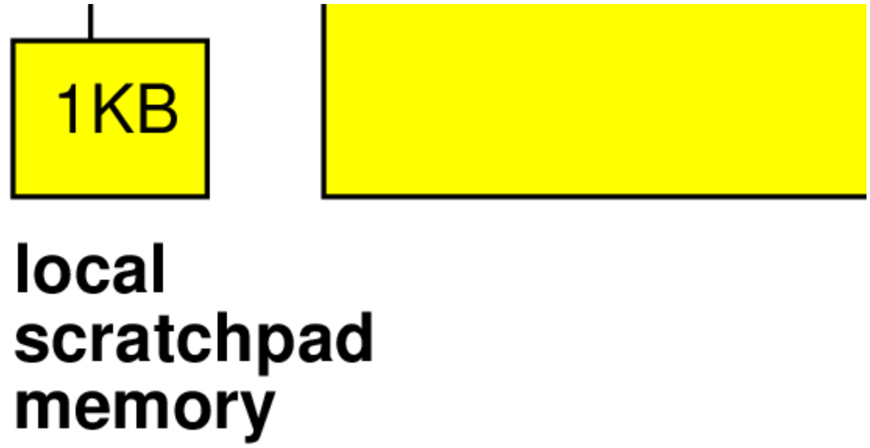
12pt Paragraph

0 words

Consider the following baseline system:

We start with a baseline, single processor system as shown.





- For simplicity throughout, we will treat non-memory indexing adds (subtracts count as adds), compares, logical operations (&&, ||,|,^&), min, max, divides, and multplies as the only compute operations. We'll assume the other operations take negligible time or can be run in parallel (ILP) with the listed compute and memory operations. (Some consequences: You may ignore loop and conditional overheads in processor runtime estimates; you may ignore computations in array indicies.)
- Baseline processor can execute one operation (as defined previous bullet) per cycle and runs at 1 GHz.
- Reads from and writes to the 1 MB main memory issue in one cycle, but require 5 cycles of latency (including issue) to get the first 64b result; memory can supply one 64b read or write each cycle. Reads larger than 64b return 64b per cycle following the first result.
- Up to 64b reads from and writes to the 1 KB scratchpad memory take 1 cycle.
- By default, all arrays live in the main memory and all array references are to main memory.
- Assume non-array variables live in registers.
- Assume all additions are associative. Max and min are associative.
- A lookup in a small memory (1KB or small) can complete in 1ns.
- A write to the pipeline accelerator above can be performed in one cycle.

Question 5

5 pts

Estimate the throughput in cycles per frame for loop A running on the baseline processor.

Question 6

5 pts

Explain your throughput answer above.

HTML Editor

12pt

Paragraph

0 words

Question 7

4 pts

Where is the bottleneck in throughput processing frames?

- Loop A compute
- Loop A memory
- Loop E compute
- Loop E memory

Question 8

4 pts

What is the Amdahl's Law speedup if you were to accelerate the bottleneck identified in the previous question? Support your answer with calculations.

HTML Editor

12pt Paragraph

0 words

Question 9

4 pts

What is the smallest granularity that you can profitably stream data between Loop A and Loop E?

- ◯ Entire tmp[] (all NFRAMES words in tmp[], each of which is a 64b word)
- ◯ single 64b word
- ◯ no streaming possible

Question 10

10 pts

Use the scratchpad memory to accelerate memory operations in Loop A.

Indicate which data you place in the scratchpad.

Provide code or other clear description of how you modify the provided code for Loop A to exploit the scratchpad memory.

Use part of this box to provide justification to the numerical answer in the next question.

HTML Editor

12pt Paragraph

0 words

Question 11 4 pts

For your revised code in the previous question, what is the throughput in cycles per frame for the revised implementation of Loop A?

Question 12

14 pts

Classify each loop as sequential, reduce, or data parallel:

- Loop A [Select]
- Loop B [Select]
- Loop C [Select]
- Loop D [Select]
- Loop E [Select]
- Loop F [Select]
- Loop G [Select]

Question 13

5 pts

What is the cycle-bound II (unlimited hardware) for loop A (assuming no bottleneck on input frames or output tmp)?

Question 14

5 pts

What is the latency bound (unlimited hardware) for loop A executing all NFRAME frames (assuming no bottleneck on input frames or output tmp)?

Question 15

5 pts

Support your numeric answers to the previous two questions.

HTML Editor

12pt Paragraph

0 words

Question 16

9 pts

Describe a VLIW architecture (types of operators and numbers of each, custom memories (memory ports) as needed) for executing Loop A that has a Resource Bound throughput (cycles per frame) that is the same throughput as the pipeline for Loop E such that Loop A is no longer the bottleneck. For simplicity, you may assume a monolithic, multi-ported register file. Try to identify an architecture with minimum hardware achieving the target resource bound. Provide description and calculations to support your answer.

HTML Editor

12pt Paragraph

0 words

Question 17

5 pts

Given the resources identified in the previous question, how close to the target resource bound will a scheduled computation achieve? Explain your reasoning.

HTML Editor

12pt Paragraph

0 words

Quiz saved at 9:49am

Submit Quiz