Final Exam

- In-person exam
  - A longer quiz covers multiple topics.
  - The final exam will be at 1:30 pm Friday, Dec. 17 @ DIAM 122.
  - The exam should take you no more than 2 hours, but you have 3 hours for it.
  - Close notes and close books. You can bring one letter size cheatsheet and a calculator. No other electronic device.
  - It’s your responsibility to make your answers readable. All answers should be brief but clear. Make sure your answers are neat if you wish full credit. I will not take off points for trivial errors, such as misspelling. However, I will take off points if your answers are messy and difficult to read.
  - You are expected to take the final exam individually.

Wrapping up …

- Binary/Decimal
  - conversion between them
  - 2’ complement

<table>
<thead>
<tr>
<th>$-2^7$</th>
<th>$2^6$</th>
<th>$2^5$</th>
<th>$2^4$</th>
<th>$2^3$</th>
<th>$2^2$</th>
<th>$2^1$</th>
<th>$2^0$</th>
</tr>
</thead>
</table>

- Digital logic
  - gates (AND, OR, NOT, NAND, NOR, XOR), their uses and implementations
  - circuit design (k-map)
    - truth table from question specification, k-map (00, 01, 11, 10), sum of produce
    - HW1 and Q1
  - useful combinational circuits: MUX, DEMUX, and Decoder.
• VHDL
  • sequential circuits: flipflops, registers
  • difference between combinational circuits and sequential circuits
  • state machines
  • programmable circuits

- Memory
  • memory hierarchy (registers, cache, main memory)
  • mapping functions: direct, associative, set associative; addressable unit: word, byte
    - HW5, Q5
  • average access-time, hit-rate
    - hit rate: the ratio that the accessed word is found in the faster memory
    - The hit rate of L1 cache is 70% and its access time is 1 ns. The hit rate for the L-2 cache is 80% and its access time is 5 ns. The access time of main memory is 10 ns. What is the average access time.
      - \[0.7 \times 1 + 0.3 \times [0.8 \times (1 + 5) + 0.2 \times (1 + 5 + 10)] = 0.7 + 2.4 = 3.1 \text{ ns}\]

- ISA
  • stack architectures
    - both operands of ALU popped from the top of the stack, result of ALU push into the top of the stack

- function call and stack

```
PUSH A # push value at location A to the top of the stack
PUSH B # push value at location B to the top of the stack
ADD # add the top two elements of the stack and save the result back to the top of the stack
POP C # store the value at the top of the stack (which is the result) to location C
```

- addressing modes (immediate, direct, indirect, register direct, register indirect, displacement, advantages/disadvantages)
Assembly language
  - the benefits of assembly language
  - two-pass assembler
  - call stack (layout, how it works)
  - assembly program (Examples on the notes)

Pipeline and superscalar
  - pipeline improves the throughput of a processor
  - superscalar supports multiple pipelines in a processor
  - dependencies (RAW, WAW, WAR), possible way to address WAW and WAR dependencies (register renaming)
  - instruction issue policies (out-of-order issue, out-of-order completion)
  - HW9, Q9