**Flip Flops­ Counters­ Sequential Circuits**

[ ] Design a Full Adder/Subtractor as a Sequential Circuit

[ ] Describe conceptually what a flip flop is

[ ] Describe what registers are made out of

[ ] Describe which (3) operations we’d like a memory unit to perform

[ ] Write the Excitation Tables and Circuit Diagrams for SR Latch, SR Latch with control input, D Latch, D Flip Flop

[ ] Write the Excitation Tables for T Flip Flop, JK Flip Flop

[ ] Describe the progression from SR Latch to D Flip Flop discussed in class

**Microprocessor, PLA, ROM, Register Transfer, HDL**

[ ] Describe what a microprocessor is

[ ] List the (4) types of Programmable Logic Devices

[ ] List the Levels of Integration for ICs

[ ] List the (5) Basic Steps for Executing a Typical Instruction by a Processor

[ ] Describe the function of the Program Counter

[ ] Describe what Register Transfer Operations are

[ ] Describe what counters are used for

[ ] Design a 3-bit Counter that counts from 0 to 7 with D, T, and JK Flip Flops (three separate designs)

**Processor & Control Design, Computer Classifications**

[ ] List Flynn’s (4) Computer Classifications and some examples, characteristics, and a diagram of each

[ ] List the (2) Computer Classifications Based on Memory and draw a diagram for each

[ ] Define a bus and list its characteristics

[ ] Define diameter

[ ] Find the max of all numbers with a bus (SIMD)

[ ] Design a CPU at the block diagram level

[ ] Describe what components are needed for implementing the (5) Basic Steps for Executing a Typical Instruction by a Processor

[ ] Define a control unit

**Instruction Set Architecture, Operation Cycle, Addressing**

[ ] Define Instruction Set Architecture

[ ] Trace the hypothetical Assembly Program with N: DATAWORD 10 and the given memory contents. Show contents of all registers.

[ ] List the (3) ISA Classifications by Operand Addressing and give an example of each

[ ] List the (3) Addressing Modes and the symbols used to indicate each mode in assembly language

[ ] List (5) Types of Instructions

[ ] Describe what shifting a binary number left 1 bit or right 1 bit does

**Pipelining vs. Multiprocessing, RISC vs. CISC**

[ ] Define Multiprocessing and discuss its drawbacks

[ ] Find the speedup of Multiprocessing for 50 research projects (4 steps x 1 day)

[ ] Define Pipelining and describe its key features

[ ] Find the speedup of Pipelining as compared to Sequential for n tasks where each task takes k units of time

[ ] Define pipeline flushing time

[ ] Define throughput

[ ] Define super scalar processing

[ ] Describe the characteristics of RISC

[ ] Describe the characteristics of CISC

[ ] Give an example of RISC vs CISC in terms of instructions

[ ] RISC vs CISC which is better and why?

[ ] Where does Pipelining fall in Flynn’s Computer Classifications?

[ ] Describe in your own words what is pipelining

[ ] Give an example of pipelining in terms of the (5) Basic Steps for Executing a Typical Instruction by a Processor

**Interrupts, DMA, Memory Systems, Cache, RAM, DISC**

[ ] Show how you could do matrix multiplication on a chip with a n x n array of processors. Each processor has a multiplier, adder, and registers. Also find the speedup. (SIMD) (Example of Pipelining and Multiprocessing)

[ ] Draw the (3) Levels of the Hierarchical Memory Structure and describe the differences in size, speed, and price/byte between the levels.

[ ] Discuss the fridge analogy for fetching data from memory

[ ] Define unit of access

[ ] Define Spatial Locality

[ ] Define Temporal Locality

[ ] Describe the (3) Placement Strategies for bringing data from secondary memory to main memory (when there is room).

[ ] Define pages

[ ] Define segmentation

[ ] List some of the (many) Replacement Strategies for bringing data from secondary memory to main memory (there is no room, decide who to replace).

[ ] List the (3) Cache Mapping Techniques for bringing data from main memory to cache and give the function, draw a diagram

[ ] Why are mapping techniques used for cache instead of the placement/replacement strategies?

[ ] Describe the speed of I/O

[ ] Describe the function of an interrupt and also what the processor must save when switching tasks

[ ] Define Direct Memory Access

**Midterm Exam Questions**

1. Full Adder (10)
   1. Show sum of products and product of sums for S, C
   2. Show K-map minimization for S, C
   3. Show sum of products is equal to product of sums for S, C
2. Design a circuit that can add -18 and 26 in Two’s Complement format
   1. Show Full Adder gate level diagram
   2. Show Ripple Carry Adder diagram
   3. Show how to make Ripple Carry Adder do subtraction
   4. Use correct number of Full Adders (find number of bits needed to represent -18 and 26 in binary – 6?)
   5. Show time complexity, space complexity for design
   6. Show result of adding -18 + 26 in Two’s Complement, BCD, Hexadecimal Two’s Complement (had to know zero fill vs sign extension), Sign Magnitude
3. Show how to algorithmically design a Comparator
   1. Show truth table, gates for =, >, and < for one bit. Write Boolean expression for 3 bits using algorithm. Show gate level diagram.
   2. Show NAND implementation
   3. Show NOR implementation
   4. Show this design on a MUX
      1. Show how to map a truth table for a design onto a MUX
      2. Recognize that this is a 16x1 MUX and draw diagram showing that it’s made out of 2x1 MUX’s
      3. Show how a 2x1 MUX is implemented (truth table and gate level diagram)
4. Misc Questions
   1. Show the 3 components of a computer
   2. Describe what you’ve learned in the course so far
   3. Describe what an Decoder is
   4. Describe what an Encoder is
   5. Describe what a Demultiplexer is
   6. Show what universal gates are (AND, OR, NOT)
   7. Prove that NAND is universal
   8. Prove that NOR is universal