

EECS 370

Discussion 3

Week of 1/30 – 2/5

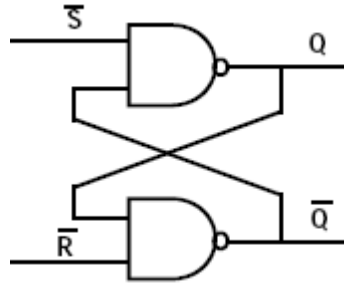
- I) Project 1
 - a. Make sure you have no newlines at the end of your assembly programs!
 - i. Check with vim
 - ii. Dos2unix
- II) Symbol/relocation tables
 - a. Symbol tables (key concept: **public entities**)
 - i. Binds public entities; allows linker to use this file's entities elsewhere, and informs linker what entities this file requires
 - ii. All globals
 - iii. All function names and calls
 - iv. All unresolved labels and names
 - b. Relocation tables (key concept: **absolute addresses**)
 - i. Many references must change if parts of the program are rearranged in memory
 - ii. Question: Do local variables rely on absolute addresses?
 - iii. References to globals
 - iv. References to functions
 - v. Note: unresolved tokens must be global in a valid program!
 - c. Code example:
 - 1. int a;
 - 2. void foo(int b)
 - 3. {
 - 4. x = b;
 - 5. printf(“%d\n”, x);
 - 6. a = 15;
 - 7. return;
 - 8. }
 - d. What symbols go in the symbol table? What lines go in the relocation table?
 - e. Linking two object files
 - i. Combined text size = sum of individual text sizes
 - ii. Likewise for data size
 - iii. Put the data in the data segment, check relocation table for references that need updating, and update them with the new addresses
- III) De Morgan's Laws

- a. $\sim(P \& Q) == (\sim P) | (\sim Q)$
- b. $\sim(P | Q) == (\sim P) \& (\sim Q)$
- c. Think about how to implement an OR gate with a NAND gate!

IV) Cascaded MUXs

- a. Can build bigger MUXs using smaller MUXs
- b. Build a 4:1 MUX using three 2:1 MUXs
- c. Use 2-level mux tree
- d. Each selector bit multiplexes one **level**

V) SRAM



- a.
- b. If R is 1 and S is 0, $Q = 1$
- c. If R is 0 and S is 1, $Q = 0$
- d. If both 1, state is retained
- e. If both 0, restricted combination (logic mismatch: $Q = !Q$)
- f. If both are 0, and jump to 1...?!