

- (3.1) Why is the program counter a *pointer* and not a *counter*?
- (3.2) Explain the function of the following registers in a CPU: PC, MAR, MBR, IR
- (3.3) For each of the following 6-bit operations, calculate the values of C, Z, V, N
- (3.10) Why does ARM provide a reverse subtract instruction?
- (3.17) ARM uses 12-bit literal. Compare and contrast the 8-bit format and 4-bit alignment vs straight 12-bit literal.
- (3.18) Write one or more ARM instructions that will clear bits 20 to 25 inclusive in register r0. All other bits of r0 should remain unchanged.
- (3.19) Swap contents of r0 and r1 without using any other registers or memory storage.
- (3.25) What is the binary encoding of the following instructions? A. STRB r1, [r2] B. LDR r3, [r4,r5] C. LDR r3,[r4],r5 D. LDR r3, [r4,#-6]
- (3.39) Write ARM assembly that scans a null terminated string and copies the string from a source pointed to by r0 to a destination pointed to by r1
- (3.51) Is it possible to have  $n$ -input AND, OR, and NOR gates where  $n > 2$ ? Explain your answer with a truth table.