Sean Penney and Paul Atkinson
April 21, 2014
CS 472 HW 3

(3.1) The program counter is not a counter. The PC holds, or points to, the memory address of the next instruction to be executed.

(3.2) a. PC points to the memory address of the next instruction to be executed.

b. MAR stores the address that is being accessed by read/write operations.

c. MBR is the memory buffer register, which holds data that has been read from main memory or will be written to main memory.

d. The IR (instruction register) holds the instruction currently being executed.

(3.3) a. C = 0, Z = 0, V = 0, N = 0

b. C = 1, Z = 1, V = 0, N = 0

c. C = 0, Z = 0, V = 0, N = 0

d. C = 1, Z = 0, V = 0, N = 0

e. C = 0, Z = 0, V = 0, N = 1

f. C = 1, Z = 0, V = 0, N = 1

(3.10) RSB (Reverse Subtract) exists because there are a wide range of options for Operand2. Operand2 can be either a constant or a register with optional shift.

(3.17) Using an 8-bit format for the integer and a 4-bit alignment field allows for a larger range of values. The disadvantage to this mechanism is that there are gaps in the range of values.

(3.18) AND r0, 0xFE0FFFFF

This will take all the bits from 20-25 and set them to zero and it will leave the rest alone.

(3.19) Using the XOR swap algorithm:

EOR r0, r0, r1

EOR r1, r1, r0

EOR r0, r0, r1

(3.25) a. 11100101110000010000100000000000

b. 11100111101101000001000000000101

c. 11100110100101000011000000000101

d. 11100101001101000011000000000110

(3.39) LOOP LDRB r2, [r0], #1 ; get address of next character

STRB, r2, [r1], #1 ; store contents at r2 in r1

TEQ r2, #0 ; check if at end of string

BRNE LOOP ; if not at end of string, go back to loop

(3.51)

```
                AREA palindrome, CODE, READONLY
                ENTRY

start    LDR r1, =string_begin
                LDR r2, =string_end
                BL pal
```

```
stop     B stop

pal            LDRB r3, [r1], #1       ; get left hand character
               LDRB r4, [r2], #-1      ; get right hand character
               CMP r3, r4                        ; compare the ends of the string
               BNE notpal                        ; if different then fail
               SUBS r3, r2, r1         ; get difference between pointers
               BEQ waspal                        ; if same then exit with palindrome fou
               BMI waspal                        ; if left pointer past right then palin
               B pal                             ; REPEAT
waspal   MOV r0, #0x1            ; r0 = 1 = success flag
               MOV pc, lr                        ; return
notpal   MOV r0, #0x0            ; r0 = 0 = fail flag
               MOV pc, lr                        ; return

               END
```