Rick Sullivan
Professor Lewis
COEN 20
20 May 2013

Homework #7

1. c) DMA
2. c) DMA
3. c) DMA
4. c) Maximum memory bandwidth
5. (a) 2 clock cycles
   (b) 2 clock cycles
   (c) 1 clock cycle
   (d) 5 clock cycles
   (e) 4 clock cycles
6. (a) 1 byte from memory
   (b) 4 bytes from memory
   (c) 0 bytes
   (d) 16 bytes to memory
   (e) 4 bytes to memory
7. a, b, c, and d
8. a
9. a, b, and c
10. a) R0, c) R12, d) LR, and e) PSW
11. c) The NVIC
12. a) A hard-coded constant built into the CPU
13. False
14. b) Tail-chaining
15. c) Late arrival processing
16. b) The starting address of the vector table is stored in the vector table offset register
17. (a) If the second interrupt is of **lower** priority, then only after the ISR returns.
    (b) Immediately if the second interrupt is of **higher** priority.
18. Registers R0-R3, R12, LR, Return Address, and PSR.
19. Interrupt-driven I/O is slowed by the overhead of saving and restoring the machine state.
20. (a) This case would be the best case latency of 340nsec.

(b) The latency depends on whether the new ISR has a higher or lower priority than the currently executing ISRs.

(c) DMA has a latency of approximately 60nsec.

(d) The latency of the polled waiting loop is unpredictable.

21.
```
void move(uint8_t *from, uint8_t *to)
{
    for(int byte = 0; byte < 1000; byte++)
    {
        *to = *from;
        to++;
        from++;
    }
}

In assembly:

        ; R0 has &from
        ; R1 has &to
move:

        LDR R3, =0          ; byte counter
Start:  CMP R3, #250        ; It's faster to do 4 bytes per loop,
        BHE Return          ;    so we only need to loop 250 times
        LDR R2, [R0], #1    ; Get next byte and update pointer
        STR R2, [R1], #1    ; Store byte and update pointer
        ADD R3, R3, #1      ; Increment counter.
        B   Start
Return: BX  LR
```

Each loop should take approximately 8 clock cycles, and it will execute 250 times. Therefore, it will take 2,000 clock cycles (+ some negligible cycles outside of the loop). At 20nsec per cycle, this should take about 40,000 nsec.