Problem 1: (20 points)

Assume an A/D converter is supplying samples at 44.1 kHz and is using interrupts to preempt the application routine. Assume a 20 MHz ARM processor that executes .9 instructions per cycle.

a) **(10 pts) If the interrupt handler executes 100 instructions obtaining the sample and passing it onto the application routine, how many application code instructions can be executed between samples?**

1/44100 Hz = 22676 ns
20 MHz = 50 ns/ instr
1/.9 = 1.1111 inst/clock

1.1111 * 50 = 55.55 ns/inst
22,676 = (100 + n) * 55.55
n = 408.2 – 100 = 308

b) **(5 pts) If the application routine requires 600 instructions between samples and the interrupt handler requires 100 instructions, what is the resulting sampling frequency of the A/D converter?**

25.7 KHz

c) **(5 pts) What would the Instructions/cycle need to be for problem b) above in order to keep the 44.1KHz sampling rate.**

1.54

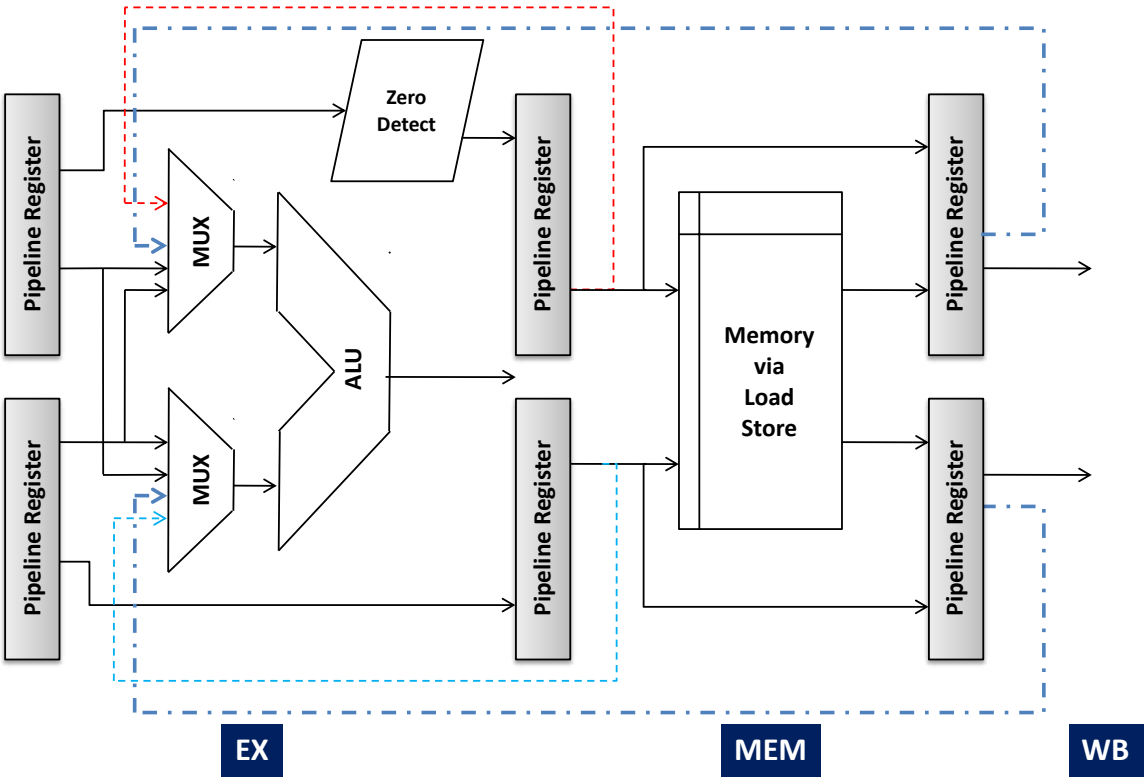
Problem 2: (20 points)

a) (10 pts) What is data forwarding? Explain in detail what control hazards are associated with implementing data forwarding.

Forwarding is a hardware-based technique (also called register bypassing or short-circuiting) used to eliminate or minimize data hazard stalls. Using forwarding hardware, the result of an instruction is copied directly from where it is produced (ALU, memory read port etc.), to where subsequent instructions need it (ALU input register, memory write port etc.)

Control hazards in data forwarding con occur when there is change of flow in the pipeline or when a bubble is inserted in the pipeline due to a stall condition.

b) (10 pts) Draw the data forwarding paths on the following diagram



Problem 3: (15 points)

a) (5 pts) What are the 3 ways to specify operands in the ARM instruction set?

1. Come from registers
2. The second operand may be a constant (immediate)
3. Shifted (scaled) register/immediate operands

b) (5 pts) What are the benefits of using a barrel shifter in front of the ALU in the Cortex A9 datapath?

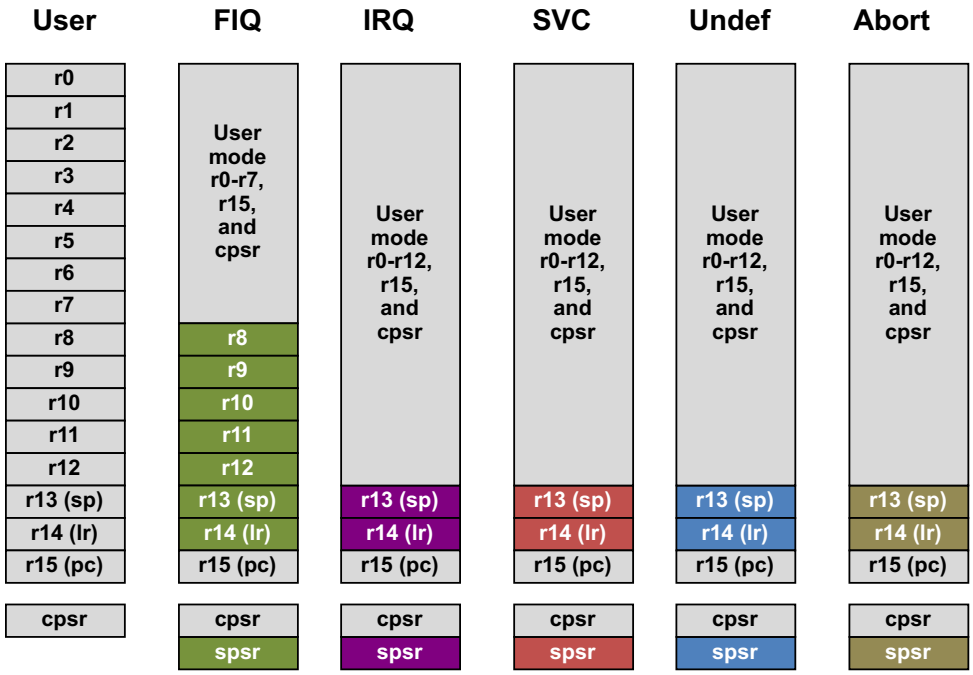
1. Scaled addressing.
2. Multiplication by an immediate value.
3. Constructing immediate values.

c) (5 pts) What are the differences between the ARM "IRQ" and "FIQ" interrupts?

1. ARM treats FIQ (Fast interrupt request) at a higher priority as compared to IRQ (interrupt request). When an IRQ is executing an FIQ can interrupt it while vice versa is not true.

2. ARM uses a dedicated banked out register for FIQ mode; register numbers R8-R12. So when an FIQ comes these registers are directly swapped with the normal working register and the CPU need not take the pain of storing these registers in the stack. So it makes it faster.

   NOTE: The FIQ does not have any more registers than any other operating mode. It simply does not have to save as many registers when there is a context switch.



Problem 4: (10 points)

a) (5 pts) Explain in detail what the three issues are with using fixed width instructions:

1. **Code density can be worse than variable width instructions.**
2. **Limited number of opcodes available**
3. **Limits the size of immediate data**
4. **Complex operations require multiple instructions**
5. **Generally limited to a LOAD-STORE memory architecture, i.e., less addressing modes**

b) (5 pts) What the three good reasons are to use fixed width instructions:

1. **Easy to decode the instruction – simplified pipelining**
2. **Minimal memory alignment issues – can determine next PC easily.**
3. **Lower power implementation**
4. **Doesn't require multi-cycle micro-coded control logic.**
5. **Can build "THUMB" like machines**
6. **Orthogonal opcode map aids code generation for HLL compilers.**

Problem 5: (15 points)

a) (5 pts) What is the average *Throughput* of a pipelined machine with a CPI=1?

1 per clock cycle

b) (5 pts) What is the average *Latency* of a 6-stage pipelined machine with a IPC=2.3?

6 clock cycles assuming that the machine cannot retire more than one instruction a cycle. High IPC machines are designed to retire more than one instruction a cycle and have multiple execution units to support execution of multiple instructions.

c) (5 pts) What are the 3 sources of interrupt latency in the Linux Kernel?

1. **Critical sections protected by spinlocks, or other section in which interrupts are explicitly disabled will delay the beginning of the execution of the interrupt handler**
2. **Equal or Higher priority interrupt handlers are executing**
3. **Registration of the hardware interrupt in the kernel due to synchronization logic gates, context saving, software, etc.**

Problem 6: (10 points)

a) (5 pts) Explain in some detail what are the 3 pros for using Kernel Modules:

1. **Kernel Modules can be compiled and dynamically linked into kernel address space after the kernel has been compiled.**
2. **Can be dynamically unloaded when not needed.**
3. **Keeps core kernel "footprint" small - used to "extend" functionality of kernel**
4. **Kernel modules run in kernel space**
    a. **Execute in the supervisor mode**
    b. **Everything is allowed**
    c. **Share the same address space**

b) (5 pts) Explain in some detail what are the 3 cons of using Kernel Modules:

1. **Generally, KMs need to be recompiled if the kernel image changes.**
    a. **Need up-to-date Linux Header files – not always available**
2. **Runs with supervisor privileges and can cause the kernel to crash**
3. **Can be a source of security issues**
4. **Can cause interrupt latency issues for other drivers if the top-half is not "Friendly"**
5. **Harder to debug**
6. **Special considerations are needed when supporting dynamic unloading.**

Problem 7: (10 points)

a) (5 pts) Explain in some detail what are the 3 pros of conditional execution instructions in the ARM architecture:

1. **Most instruction sets only allow branches to be executed conditionally. By reusing the condition evaluation hardware, ARM effectively increases number of instructions.**
2. **Removes the need for many branches, which stall the pipeline (3 or more cycles to refill depending on the architecture).**
    a. **The Time penalty of not executing several conditional instructions is frequently less than overhead of the branch or subroutine call that would otherwise be needed.**
3. **Improves code density**
4. **Provides compiler more optimization opportunities.**

b) (5 pts) Explain in some detail what are the 3 cons of conditional execution instructions in the ARM architecture:

1. **Consumes 4 bits of every instruction. Frequency of usage can be low. Might be able to use the 4 bits to extend in the instruction set differently.**
2. **Useful only for replacing forward branches.**
3. **There is a limit to how many CE instructions can be used versus just branching and taking the branch penalty.**
4. **Non-executed instructions consume 1 cycle. Still have to complete cycle so as to allow fetching and decoding of the following instructions with minimal control hazard issues.**
5. **More complex code generation in the compiler.**