LSU EE 4720 Homework 2 Due: 29 February 2008

For the answers to these questions look at the ARM Architecture Reference Manual linked to the course references page, `http://www.ece.lsu.edu/ee4720/reference.html`.

Problem 1: The register fields in ARM instructions are four bits and so only 16 integer registers are accessible. The ISA manual describes ARM as having 32 integer registers, however many of them are only accessible in particular modes.

An advantage of fewer registers is that extra bits are available in the instruction encoding, for example, ARM three-register instruction formats would have three more bits available than the MIPS type R format. Where in the ARM formats do you think these bits went? In your answer give the instruction field and its purpose. There should be no equivalent in MIPS.

Problem 2: In MIPS an arbitrary 32-bit constant can be loaded into a register using a `lui` followed by an `ori`. In ARM the immediate field for data-processing (integer) instructions is only 8 bits.

(*a*) Show ARM code to put an arbitrary 32-bit constant into a register without using a load instruction. Use as few instructions as possible. *Hint: take advantage of ARMS shift and rotate capabilities.*

(*b*) Show how ARM can put an arbitrary constant into a register with one load instruction, whereas in MIPS two would be required. The MIPS code is shown below. Do not assume the address of the constant is **already** in a register, that would make this problem insultingly easy! *Hint: Use one of ARM's special purpose registers.*

```
.text
 lui r1, 0x1111
 lw r1, 0x2220(r1)
 # ... a few more instructions ..
 jr $ra
 nop

.data
my_32_bit_constant:  # Address: 0x11112220
.word 0x12345678
```

Problem 3: In ARM the program counter is register `r15`, and so as far as instruction encoding goes, is treated as a general-purpose register.

(*a*) Why would really keeping the program counter in the integer register file add to the cost of an implementation?

(*b*) How does the ISA manual hint that blue parts of the implementation below is what they had in mind? (Register `r15` is not stored in the register file, it will always be bypassed from the real PC.) (Note: The ARM implementation is far from complete and parts may not work.)