Homework 7: ISA

CS 200 • 10 Points Total
Due Wednesday, March 29, 2017

Assignment

Read chapter 5 of Computer Organization and Architecture and the accompanying slides and answer the following problems.

1. Suppose I have the following memory map.

ADDRESS	DATA (hex)
…	
3C6E	32
3C6F	AA
3C70	70
3C71	61
3C72	6C
3C73	53
3C74	41
3C75	62
3C76	AF
3C77	D7
3C78	57
3C79	6F
3C7A	6C
3C7B	66
3C7C	1F
3C7D	20
…	

Show the contents of a double-word (4 bytes) starting at address 0x3C70 in Big Endian Order in hexadecimal, unsigned integer decimal, and ASCII characters. (3 pts.)

0x70616C53, 1885432915, and “palS”

2. Using the same memory map, show the contents of a double-word (4 bytes) starting at address 0x3C78 in Little Endian Order in hexadecimal and ASCII characters. (2 pts.)

0x666C6F57, and “Wolf”

3. Given 16-bit instructions, is it possible to encode the following instructions using expanding opcodes? If not, prove it. If so, give an example of a working scheme (see the Chapter 5 slides, #29 – 33 for guidance). (5 pts.)

   There are 8 registers (3-bit register addresses); memory addresses are 8 bits. So we can have the following formats:

Type	Instruction Format	Instruction Layout
A	Opcode only	OOOOOOOOOOOOOOOO
B	Opcode w/ 1 register operand	OOOOOOOOOOOOORRR
C	Opcode w/ 2 register operands	OOOOOOOOOORRRRRR
D	Opcode w/ memory operand	OOOOOOOOMMMMMMMM
E	Opcode w/ 3 register operands	OOOOOOORRRRRRRRR
F	Opcode w/ 1reg and 1memory operand	OOOOORRRMMMMMMMM
G	Opcode w/ 2reg and 1memory operands	OORRRRRRMMMMMMMM

   We want 8 Type A instructions, 7 each Type B and Type C instructions, 3 Type D instructions, 6 each Type E and Type F instructions, and 3 Type G instructions. (Hint: start with the shortest opcode.) I'll start you off with Type G:

TYPE	RANGE
A	to
B	to
C	1111111100XXXXXX to 1111111111XXXXXX
D	11111100XXXXXXXX to 11111110XXXXXXXX
E	1111000XXXXXXXXX to 1111101XXXXXXXXX
F	11000XXXXXXXXXXX to 11101XXXXXXXXXXX
G	00XXXXXXXXXXXXXX to 10XXXXXXXXXXXXXX

   Type C only has room for 4 opcodes: 1111111100, 1111111101, 1111111110, and 1111111111. But we want 7 opcodes and we can't steal another bit from the operands, so it cannot be done.