Homework #2

1. a) What is the stored-program concept?
   - It is the idea that instructions & data can be stored in memory as numbers.

   b) List the main steps of instruction execution of the stored-program concept.
   - The CPU fetches the instructions / data from memory
   - The CPU does work on the data by following the instructions
   - The CPU stores the results of the work back into memory.

2. a) How many G-P registers are available? How many bits for each register? How many bits are needed to address each register?
   - In MIPS architecture, there are 32 registers.
   - Each register is 32-bits wide.
   - you will need 1024 bits to address all 32, 32-bit registers

   b) How many bits are used for memory address? Is the memory byte-addressed or word-addressed? What is the largest memory size that we can address directly?
   - 32-bits are used for memory addressing.
   - The memory is byte-addressed in MIPS
   - 2^{32} or about 4 GBytes is the largest size we can address in memory.

   c) If 6-bits are used for the opcode field, How many different operations can be represented? How does MIPS provide extra operations beyond the 6-bit opcode?
   - 2^6 or 64 operations can be represented.
   - The last field, function, can specify a variant of the operation in the opcode field.

3. convert the following instructions into binary.

or $R12, $R15, $R16

0 15 16 12 0 25hex

000000 01111 10000 01100 00000 100101

sw $R2, 0xA012($R20)

0x2B 20 2 0xA012

101011 10100 00010 1010000000010010

4. convet the bit pattern into MIPS instruction.

0000 0001 0110 0110 0100 1000 0010 0000

0 11 6 9 0 0x20

add $R9, $R11, $R6 — add $t1, $t11, $a2

1000 1101 0100 1010 0000 0000 0001 1100

2 3 8 11 0x001C

lw $R11, 0x001C($R8) — lw $t3, 0x001C($t0)

5. Decode the MIPS instruction 0x360B0019

0011 0110 0000 1011 0000 0000 0001 1001

13 16 11 19

ORI $R11, $R16, 0x0019

a) what type/format of instruction is it?
- This is an I-type instruction

b) What addressing mode is being used?
- This is immediate addressing

c) write answer as a complete MIPS instruction?

6. How many different ways/locations could you store a 32-bit integer?
- There are 4 "blocks" of memory we could use where each block starts with the least significant bit as 00 for alignment.
- Only one block is available that is not already filled with data. That block starts with the bit-address of 1000.