

**Question 1:** (20 points)

You are participating in the Computing Science Industrial Internship Program, and your placement is with *Tiny Inc.*, a company that produces *TinyProc*— a new processor developed for the automobile industry. All instructions in *TinyProc* have 16 bits. *TinyProc* also works with 16-bit addresses. The format of a branch instruction in *TinyProc* is as shown below:

|        |    |    |    |    |   |         |   |
|--------|----|----|----|----|---|---------|---|
| 15     | 13 | 12 | 10 | 9  | 7 | 6       | 0 |
| Opcode |    | rs |    | rt |   | address |   |

Where **rs** and **rt** specify the source and target registers for the branch instruction, respectively. The address of the target of a branch instruction is computed using the same mechanism used in the MIPS processor, but the increment of the PC and the shift left have to be adjusted for a 16-bit address machine: first the Program Counter (PC) is incremented by two, then the bitfield **address** of the branch instruction is shifted left by one, sign-extended to sixteen bits, and added to the incremented PC. Based on this information, answer the following questions.

- (5 points) What is the binary representation of the **address** field of a branch instruction that results in the largest jump backward and on the largest jump forward in *TinyProc*?
- (5 points) The *range* of a branch instruction is the address distance between the target of a branch instruction and the branch instruction itself. For example if a branch instruction is at the address  $0x0010$  and the target is at address  $0x0020$ , then the range of this branch instruction is  $0x0010 = 16_{10}$ . What is the maximum range of a branch instruction, expressed as a decimal number in *TinyProc*?
- (5 points) How many registers does *TinyProc* have?
- (5 points) The instruction `beq $0, $1, SKIP` branches to the instruction at the label `SKIP` if the value in register `$0` and in register `$1` are the same. In *TinyProc*, the **Opcode** for a `beq` instruction is `010`. Assume that this instruction is at address  $0xFC00$ , and that the label `SKIP` is at address  $0xFB82$ . What is the binary representation of this instruction expressed in Hexadecimal?