►Solution◄

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.

a. (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*?

Solution:
the largest jump backward: 100 0000
the largest jump forward: 011 1111

b. (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 is at `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*?

Solution:

$$\begin{aligned}
range &= |PC - FurthestTarget| \\
FurthestTargetBack &= PC + 2 + \texttt{0xFF80} \\
rangeBack &= |PC - (PC + \texttt{0xFF82})| \\
rangeBack &= \texttt{0xFF82} = 1111\ 1111\ 1000\ 0010 \\
rangeBack &= 0000\ 0000\ 0111\ 1110 = 2^7 - 2^1 = 128 - 2 = 126 \\
FurthestTargetForward &= PC + 2 + \texttt{0x007E} \\
rangeForward &= |PC - (PC + \texttt{0x0080})| = \texttt{0x0080} = 2^7 = 128
\end{aligned}$$

Therefore, the maximum range for a branch instruction is **128 instructions**.

c. (5 points) How many registers does *TinyProc* have?

Solution: Each register field has three bits, therefore there are eight possible combinations for this field: 000, 001, 010, 011, 100, 101, 110 and 111. Thus, *TinyProc* has **8** registers.

d. (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?

Solution:

$$\begin{aligned}(PC+2)+dist &= target\\ dist &= target-(PC+2)\\ dist &= target+(-(PC+2))\end{aligned}$$

(PC+2): `0xFC02` =	1111	1100	0000	0010
/(PC+2) =	0000	0011	1111	1101
-(PC+2) =	0000	0011	1111	1110
target: `0xFB82` =	1111	1011	1000	0010
dist = target + (-(PC+2)) =	1111	1111	1000	0000
dist >> 1 =	1111	1111	1100	0000

Therefore, the `address` field for the instruction is `100 0000`.

The rest of the instruction format is:

15 13	12 10	9 7	6 0
`010`	`000`	`001`	`100 0000`

010 000 001 100 0000 = 0100 0000 1100 0000 = `0x40C0`