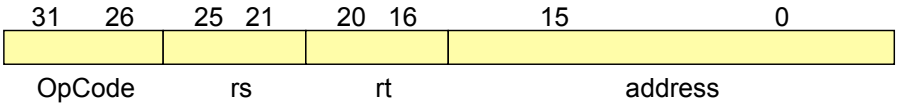
►Solution◄

Question 1: (10 points)

For the MIPS code given in Question **??**, notice the `bne` and `j` instructions at the addresses `0x0040 0010` and `0x0040 0020` respectively. What are the hexadecimal representations of these instructions? To solve this problem, you need to recall that registers `$t3` and `$zero` are mapped to register number 11 and 0, respectively. Also, you need to recall the formats and effects of branch and jump instructions. They are shown below.

a. (5 points) Branch Instruction




```
bne rs, rt, address <==> PC <-- PC + 4
                         if(rs != rt)
                         PC <-- PC + sign_extended(address << 2)
```

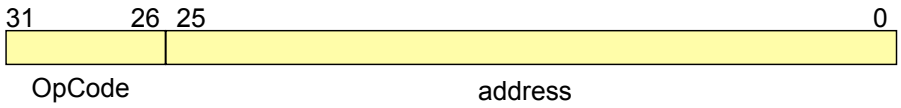
The opcode of `bne` instruction is `000101`.

> **Solution:** When the processor is executing the `bne` instruction, the PC points to the next instruction. The next instruction is at `0x0040 0014`. The target of the branch, i.e., `L2`, is 4 bytes away from this instruction. While computing the target address of a branch, the processor internally left-shifts the address specified in the branch instruction by two bits. Therefore, the address that we must specify in the `beq` instruction should be $4/4 = 1$.
>
> Thus, for the `beq` instruction, we have the values of each field as follows:
>
> $$\texttt{opcode} = 000101$$
> $$\texttt{rs} = 01011$$
> $$\texttt{rt} = 00000$$
> $$\texttt{address} = 0000000000000001$$
>
> Therefore, the binary representation of the `bne` in the given MIPS program is: `0001 0101 0110 0000 0000 0000 0000 0001`.
>
> The hexadecimal representation is: `0x 1560 0001`.

b. (5 points) Jump Instruction

```
j address <==> PC <-- PC + 4
               PC <-- concat (PC[31-28], IR[25-0]) << 2
```

The opcode of `j` instruction is `000010`.



Solution: In binary, the target address of the jump is: `0000 0000 0100 0000 0000 0000 0000 1000`. While computing the target address of the `j` instruction, the processor concatenates the bits 31-28 of the PC with the bits 25-0 of the `j` instruction and left shifts the result by 2 bits. Remember that bits 25-0 of the `j` instruction represent the address. Therefore, accounting for left shift by 2-bits, the address that must be specified in the `j` instruction is:
`00 0001 0000 0000 0000 0000 0010`.

Thus, for the `j` instruction above, we have values of each field as follows:

$$\texttt{opcode} = 000010$$
$$\texttt{address} = 00\ 0001\ 0000\ 0000\ 0000\ 0000\ 0010$$

Therefore, the binary representation of the `j` instruction in the above MIPS program is: `0000 1000 0001 0000 0000 0000 0000 0010`.

The hexadecimal representation is: `0x 0810 0002`.