1. In the instruction memory drawn below and assuming the instructions start at address 0x0, fill out what instruction memory would look like for following instructions and corresponding endian-ness.

Little Endian:

```
addi $10, $15, 7
or  $15, $6, $5
sub $6, $9, $10
```
Big Endian:

nor $6, $9, $8
and $10, $20, $14
add $5, $8, $10
2. Extend the attached datapath to support this instruction:

\[ \text{subir rt, imm32, rs} \quad \# \text{subtract register data from immediate value} \]
\[ \text{# store the result of the sum in register rt} \]
\[ \text{# uses I-type encoding} \]

\[ R[rt] = \text{imm32} - R[rs] \]

Write the values of all decoder outputs too. \textit{Recall that the ALU operations are encoded as: ADD=3'b010, SUB=3'b011, AND=3'b100, OR=3'b101, NOR=3'b110, and XOR=3'b111.}