SystemVerilog struct and union example (from ECE 571 exam)

The MIPS processor architecture adheres to the RISC philosophy in that all of the instructions are 32-bits wide. In a move towards simplicity (compared to other CPU architectures) there are only 3 different instruction formats. These three formats are called R-format, I-format and J-format and are shown in the table below:

Format	Fields						Comments
	6 bits 31..26	5 bits 25..21	5 bits 20..16	5 bits 15..11	5 bits 10..6	6 bits 5..0	All MIPS instruction are 32 bits.
R-format	opcode	rs	rt	rd	shamt	F_Code (funct)	ALU instructions except immediate, Jump Register (JR)
I-format	opcode	rs	rt	address/immediate			Load, store, immediate ALU, beq, bne
J-format	opcode	target address					Jump (J)

Define SystemVerilog packed struct(s) for the following instruction formats:

- R-format instructions

```
// R-format instructions
typedef struct packed {
        logic [5:0]         opcode;
        logic [4:0]         rs;
        logic [4:0]         rt;
        logic [4:0]         rd;
        logic [4:0]         shamt;
        logic [5:0]         funct;
} R_format_t;
```

- I-format instructions

```
// I-format instructions
typedef struct packed {
        logic [5:0]         opcode;
        logic [4:0]         rs;
        logic [4:0]         rt;
        logic [15:0]        addr_imm;
} I_format_t;
```

Format	Fields						Comments
	6 bits 31..26	5 bits 25..21	5 bits 20..16	5 bits 15..11	5 bits 10..6	6 bits 5..0	All MIPS instruction are 32 bits.
R-format	opcode	rs	rt	rd	shamt	F_Code (funct)	ALU instructions except immediate, Jump Register (JR)
I-format	opcode	rs	rt	address/immediate			Load, store, immediate ALU, beq, bne
J-format	opcode	target address					Jump (J)

- J-format instructions

```
// J-format instructions
typedef struct packed {
      logic [5:0]        opcode;
      logic [25:0]       targetAddr;
} J_format_t;
```

Define a SystemVerilog packed union that maps all of the instruction types to a single typedef called mips_instr_t. Include an additional mapping that encompasses all 32-bits of the instruction.

```
// Union of all the instruction types
typedef union packed {
      r_format_t  R;
      I_format_t  I;
      J_format_t  J;

      logic [31:0]      instr;      // whole instruction
} mips_instr_t;
```