Computation Structures — Tutorial 3

September 29, 2015

μ-code for ULg02 – User & Supervisor modes

Reminder

- ULg02 introduces the *supervisor (SVR) mode* that can execute privileged instructions.
- The current mode is encoded in the most significant bit of the PC:
  - `PC31 == 0` corresponds to *user* mode;
  - `PC31 == 1` corresponds to *SVR* mode.
- SVR mode can be activated in two ways:
  1. When using the `SVR()` β-assembly instruction.
  2. Following an interrupt.
- Jumping instructions (`JMP`, `JMPI`,...) use *absolute addressing*: they can be used to come back to user mode.
- Jumping to a privileged address (i.e., an address whose MSB is 1) from user mode is **not possible**: see the PC register circuitry.
- Branching instructions (`BR`, `BT`, `BEQ`,...) use *PC-relative addressing*: they cannot be used to come back to user mode nor to try to reach SVR mode.

Exercises

1. Provide the ULg02 supervisor micro-code for the following instruction:

```
JMPI(Ra, lit, Rc): PC <- PC + 4
                   EA <- (Reg[Ra]+SEXT(lit)) & 0xfffffc
                   Reg[Rc] <- PC
                   PC <- Mem[EA]
```

2. Provide the ULg02 user *and* supervisor micro-code for the `BT(Ra, label, Rc)` instruction. If `Reg[Ra]` is zero, it does nothing. Else, it saves the address of the instruction following `BT` then transfers execution to the address $PC + 4 \times Lit$ where Lit is computed from `label` as it is done in `BEQ`:

$$Lit = \frac{OFFSET(label) - OFFSET(CurrentInstruction)}{4} - 1$$

3. Provide the ULg02 user *and* supervisor micro-code for the `JMPB(Ra, Rb, Rc)` that behaves like `JMP(Ra, Rc)` if `Reg[Rb] >= 0` and does nothing otherwise.