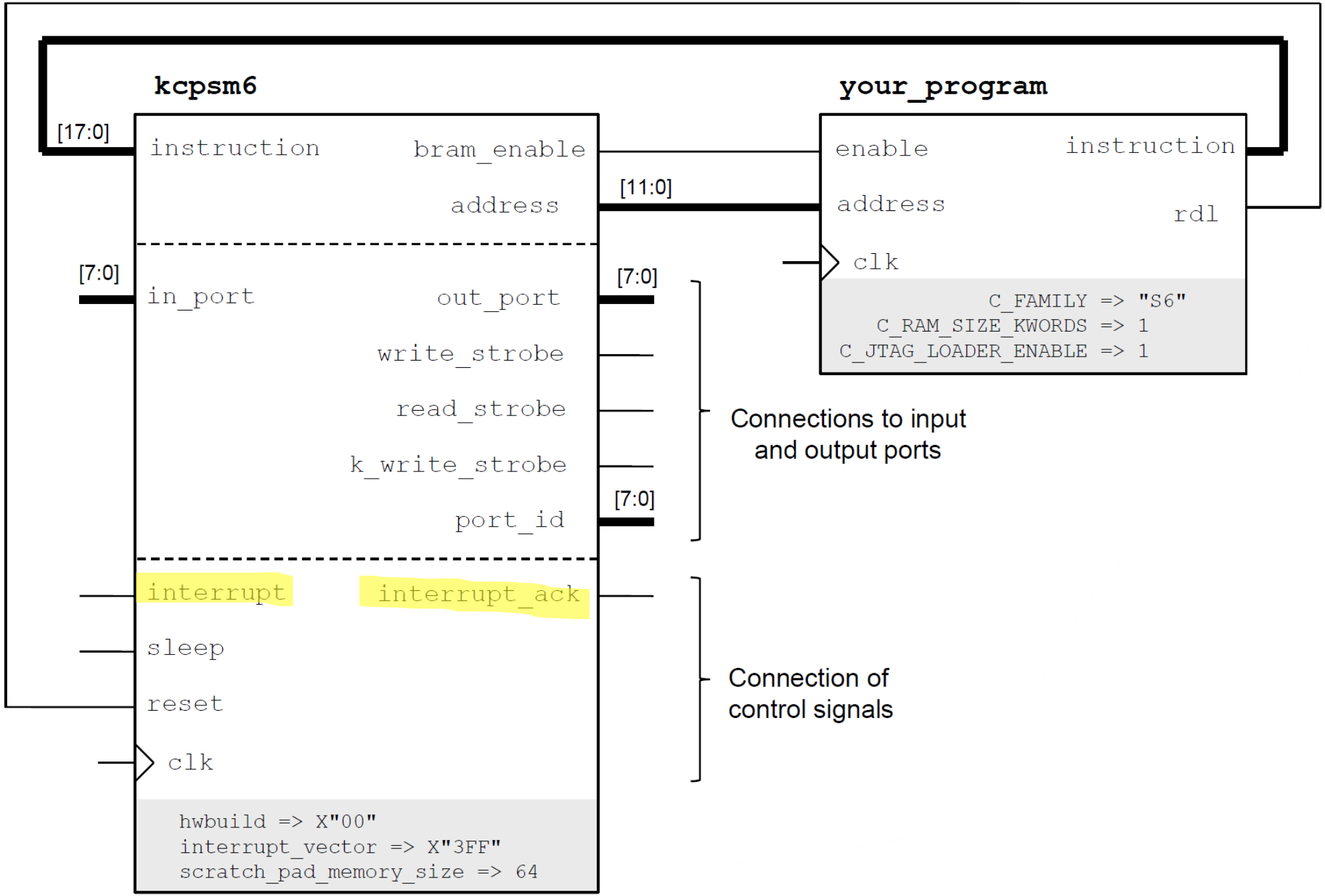
lab 06

PicoBlaze KCPSM6: interrupts

Digital design – laboratory exercises

assistant: Nejc Ilc


kcpsm6
your_program
[17:0]
instruction
bram_enable
address
[11:0]
enable
instruction
address
rdl
clk
[7:0]
in_port
out_port
write_strobe
read_strobe
k_write_strobe
port_id
[7:0]
[7:0]
C_FAMILY => "S6"
C_RAM_SIZE_KWORDS => 1
C_JTAG_LOADER_ENABLE => 1
Connections to input and output ports
interrupt
interrupt_ack
sleep
reset
clk
Connection of control signals
hwbuild => X"00"
interrupt_vector => X"3FF"
scratch_pad_memory_size => 64

Interrupt mechanism

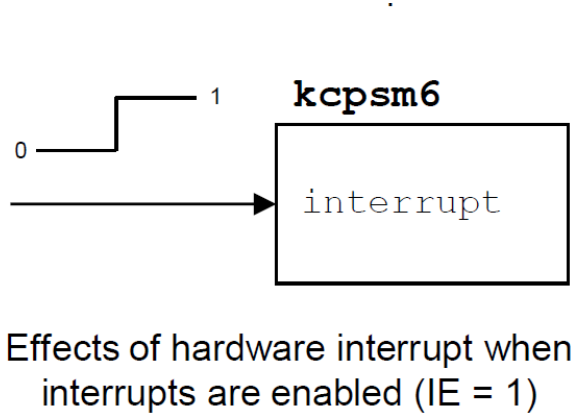
- We connect interrupt source (signal `int_request` in a template) to the input `interrupt`
  - there is an interrupt active when `interrupt` is high
  - `interrupt` should be high for at least two clock cycles
- Signal `interrupt_ack` tells us that interrupt has been accepted
  - we put `interrupt` to 0 when `interrupt_ack` is high

Example of the interrupt interface

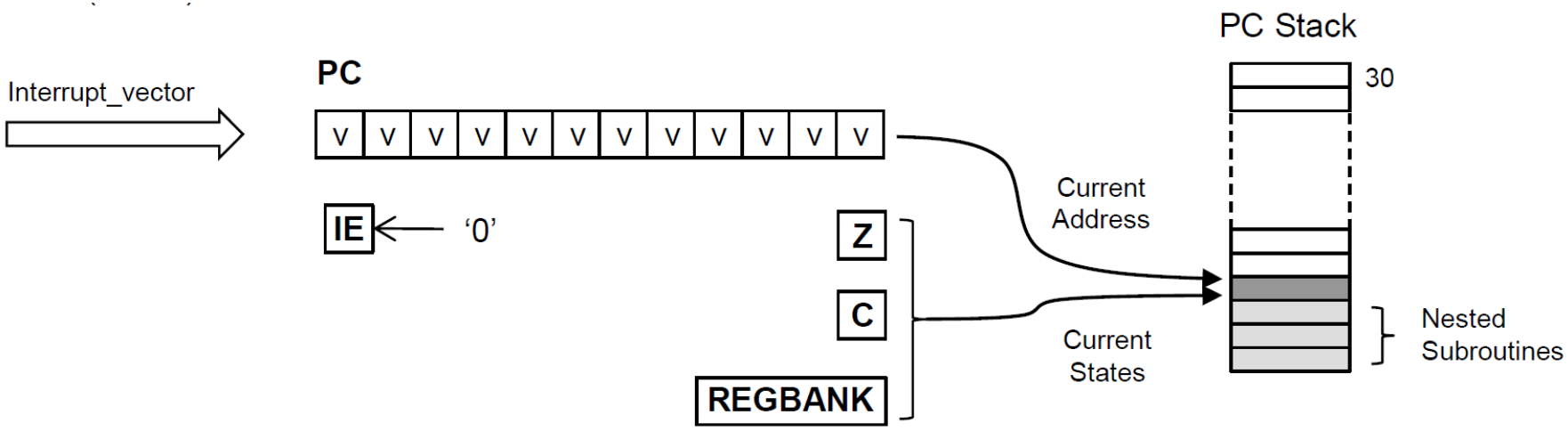
```
interrupt_control: process (clock)
begin
    if rising_edge(clock) then
        if reset = '1' then
            interrupt <= '0';
        elsif interrupt_ack = '1' then
            interrupt <= '0';
        elsif int_request = '1' then
            interrupt <= '1';
        else
            interrupt <= interrupt;
        end if;
    end if;
end process;
```

PSM instructions

- Interrupt Service Routine (ISR) has to start at address 0x3FF
  - or as specified by the `interrupt_vector`
- Interrupts are disabled by default after the reset/start of processor
- Enabling/disabling interrupts:
  `ENABLE INTERRUPT` or `DISABLE INTERRUPT`
- **Always** when returning from ISR: `RETURNI ENABLE` or `RETURNI DISABLE`



Effects of hardware interrupt when interrupts are enabled (IE = 1)

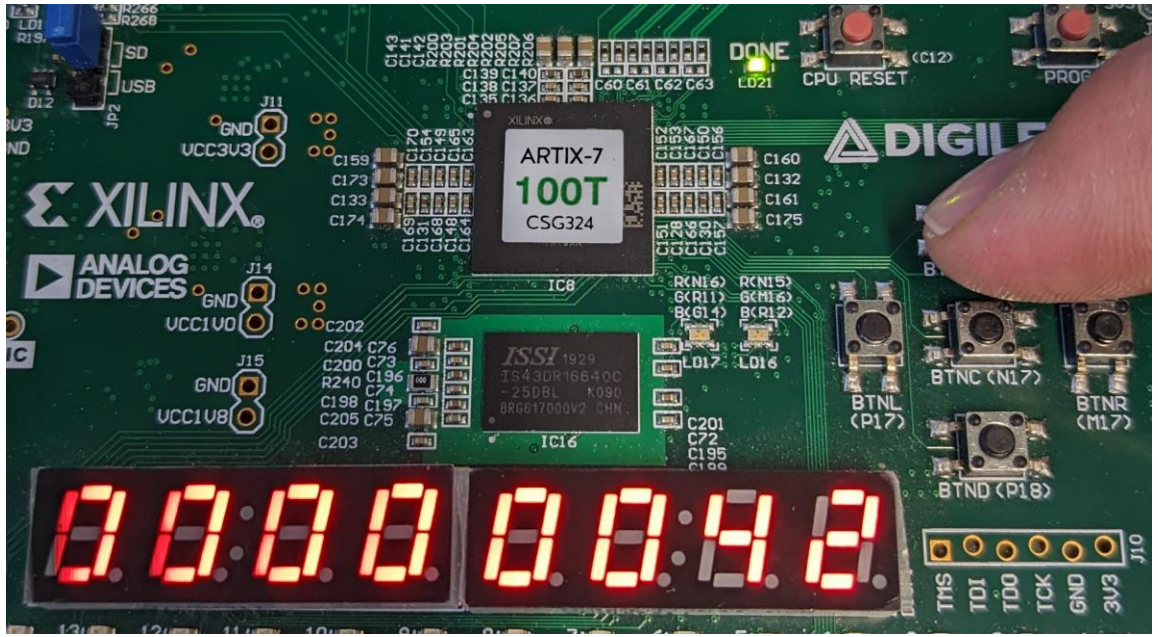


Example

```
        ADDRESS 000
        LOAD s0, 07
        OUTPUT s0, 01
        ENABLE INTERRUPT
loop:   JUMP loop
isr:    ADD s0, 01
        OUTPUT s0, 01
        RETURNI ENABLE

        ADDRESS 3FF
        JUMP isr
```

Challenge



- Pressing the top (BTNU) or bottom (BTND) button should trigger an interrupt. The ISR should change the value of the register that holds the current counter value:
  - the counter value is incremented when BTNU is pressed,
  - the counter value is decremented when BTND is pressed.
- Show the counter value on a seven-segment display.
- Reset your circuit (and PicoBlaze) with CPU_RESETN button.