SISTEMI EMBEDDED

SOPC DE10-Lite Basic Computer

Parallel port

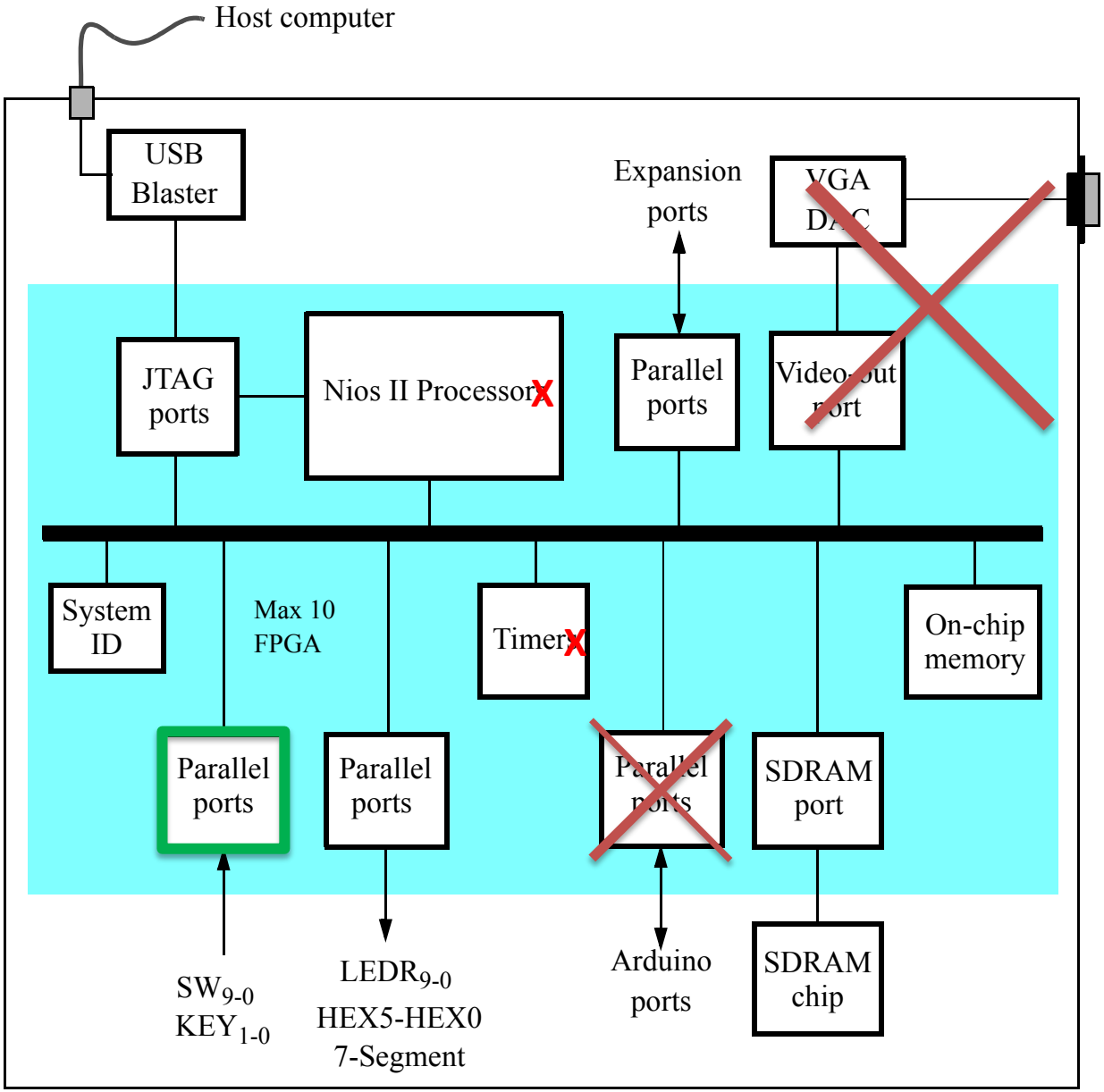
Federico Baronti

Last version: 20180307

DE10-Lite Basic Computer

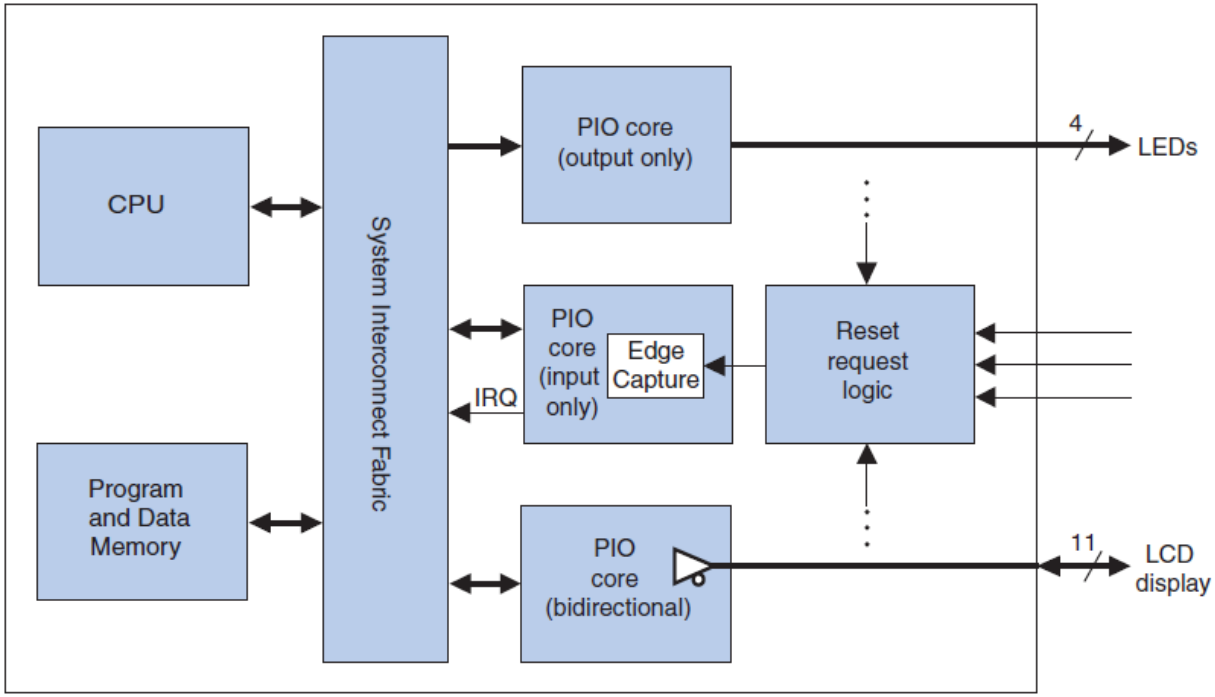
- **Simplified** version of the DE10-Lite Computer provided by Intel University Program
  - Processor: Nios II/e
  - Memory: SDRAM and On-chip memory
  - I/O:
    - Parallel ports: LEDs, HEX3_HEX0, HEX5_HEX4, Sliders_Switches, Pushbuttons, etc.
    - Other peripherals: JTAG UART, Interval_timer, sysid

DE10-Lite Basic Computer (cont.)



Parallel port (1)

- **Peripheral interface** for general purpose I/O
  - Based on Altera's PIO core customized for DE-series boards
  - Controlling LEDs, acquiring data from Switches, etc.



Parallel port (2)

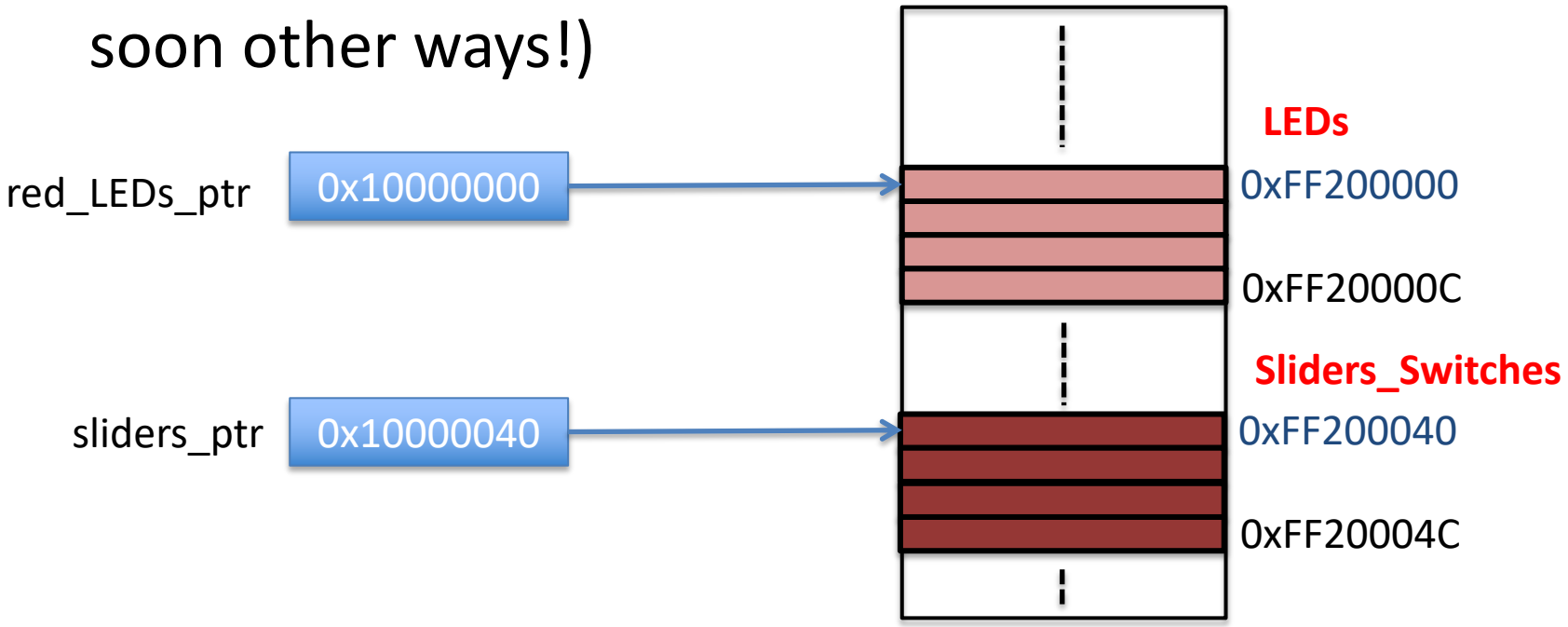
- 4x 32-bit memory-mapped registers
- *n* actual number of I/O pins

Table 2. Parallel Port register map

Offset in bytes	Register name		Read/Write	Bits $(n-1)\ldots 0$
0	`data`	Input	R	Data value currently on Parallel Port inputs.
		Output	W	New value to drive on Parallel Port outputs.
4	`direction`		R/W	Individual direction control for each I/O port. A value of 0 sets the direction to input; 1 sets the direction to output.
8	`interruptmask`		R/W	IRQ enable/disable for each input port. Setting a bit to 1 enables interrupts for the corresponding port.
12	`edgecapture`		R/W	Edge detection for each input port.

Parallel port (3)

- Managing PIO in C program:
  - Use of pointers to *unsigned int* initialized with PIO base memory address (we'll learn soon other ways!)




```
volatile unsigned int *red_LED_ptr = (unsigned int *) 0xFF200000; //red LED address
volatile unsigned int *sliders_ptr = (unsigned int *) 0xFF200040;
                                                    //SW slider switch address
```

Parallel port (4)

- Why **volatile** attribute?
  - I/O registers may change even if the program does not modify them!
    - The peripheral hardware may modify their contents
  - **Volatile** tells the compiler do not make any optimization to the code involving an object declared with the **volatile** attribute

Parallel port (5)

- Reading/Writing I/O registers:

```
*red_LED_ptr = *slider_ptr;
```

Putting into practice

- Let's start our first program with Nios II processor
  - Control the status of each DE10-Lite board red LED through the corresponding slider switch ($LEDR_i = Sw_i$)