Lab 3 Documentation

Saturday, March 9, 2019 6:42 PM

2.5.6 Parallel Ports

There are several parallel ports implemented in the FPGA that support input, output, and bidirectional transfers of data between the ARM A9 processor and I/O peripherals. As illustrated in Figure 8, each parallel port is assigned a *Base* address and contains up to four 32-bit registers. Ports that have output capability include a writable *Data* register, and ports with input capability have a readable *Data* register. Bidirectional parallel ports also include a *Direction* register that has the same bit-width as the *Data* register. Each bit in the *Data* register can be configured as an input by setting the corresponding bit in the *Direction* register to 0, or as an output by setting this bit position to 1. The *Direction* register is assigned the address *Base* + 4.

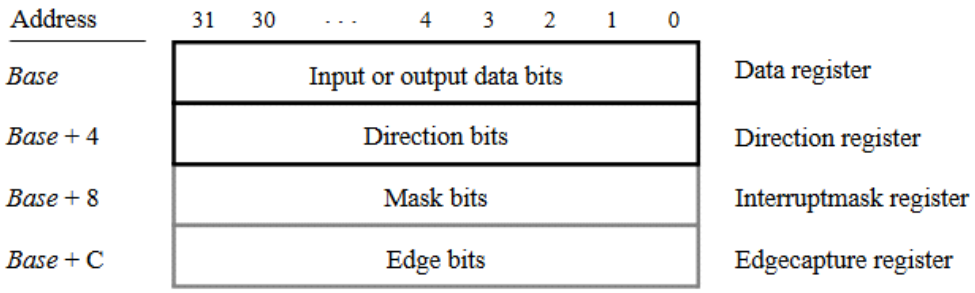

Figure 8. Parallel port registers in the DE1-SoC Computer.

Some of the parallel ports in the DE1-SoC Computer have registers at addresses *Base* + 8 and *Base* + C, as indicated in Figure 8. These registers are discussed in Section 3.

2.5.7 Red LED Parallel Port

The red lights $LEDR_{9-0}$ on the DE1-SoC board are driven by an output parallel port, as illustrated in Figure 9. The port contains a 10-bit *Data* register, which has the address 0xFF200000. This register can be written or read by the processor using word accesses, and the upper bits not used in the registers are ignored.

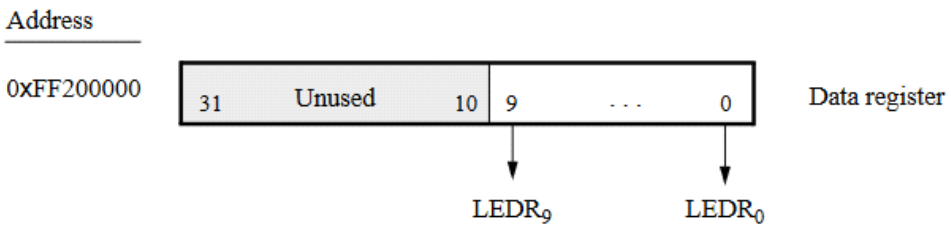

Figure 9. Output parallel port for *LEDR*.

2.5.8 7-Segment Displays Parallel Port

There are two parallel ports connected to the 7-segment displays on the DE1-SoC board, each of which comprises a 32-bit write-only *Data* register. As indicated in Figure 10, the register at address 0xFF200020 drives digits *HEX3* to *HEX0*, and the register at address 0xFF200030 drives digits *HEX5* and *HEX4*. Data can be written into these two registers, and read back, by using word operations. This data directly controls the segments of each display, according to the bit locations given in Figure 10. The locations of segments 6 to 0 in each seven-segment display on the DE1-SoC board is illustrated on the right side of the figure.

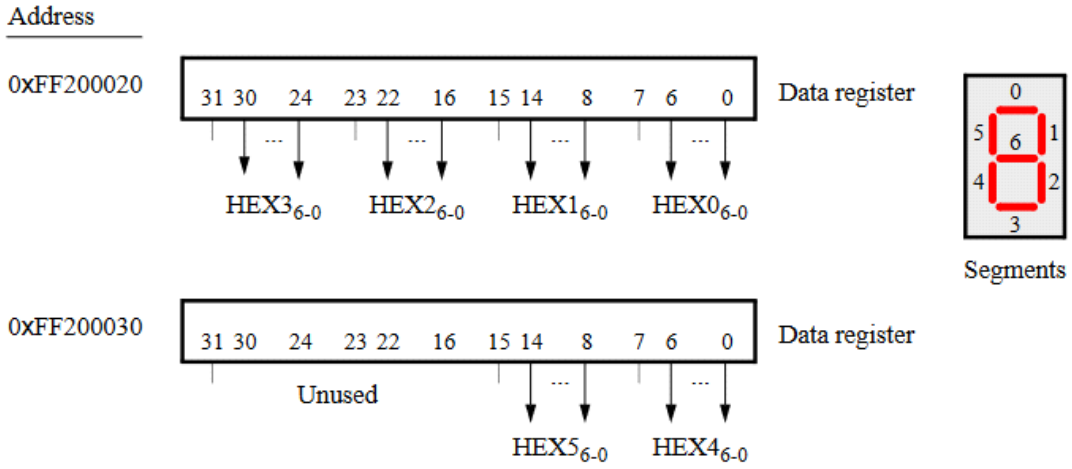

Figure 10. Bit locations for the 7-segment displays parallel ports.

2.5.9 Slider Switch Parallel Port

The SW_{9-0} slider switches on the DE1-SoC board are connected to an input parallel port. As illustrated in Figure 11, this port comprises a 10-bit read-only *Data* register, which is mapped to address 0xFF200040.

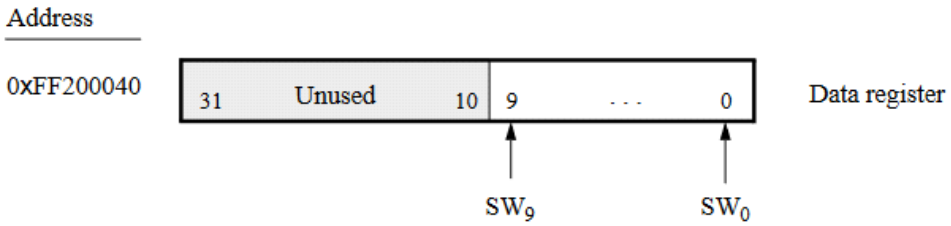

Figure 11. *Data* register in the slider switch parallel port.

2.5.10 Pushbutton Key Parallel Port

The parallel port connected to the KEY_{3-0} pushbutton switches on the DE1-SoC board comprises three 4-bit registers, as shown in Figure 12. These registers have the base address 0xFF200050 and can be accessed using word operations. The read-only *Data* register provides the values of the switches KEY_{3-0}. The other two registers shown in Figure 12, at addresses 0xFF200058 and 0xFF20005C, are discussed in Section 3.

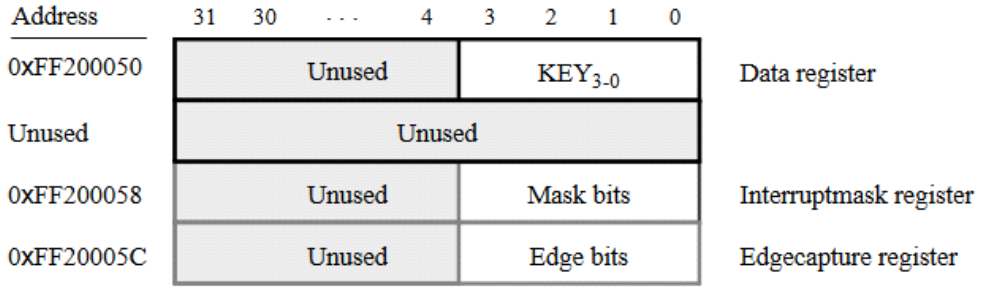

Figure 12. Registers used in the pushbutton parallel port.

3.4 Interrupts from Parallel Ports

Parallel ports implemented in the FPGA in the DE1-SoC Computer were illustrated in Figure 8, which is reproduced as Figure 21. As the figure shows, parallel ports that support interrupts include two related registers at the addresses *Base* + 8 and *Base* + C. The *Interruptmask* register, which has the address *Base* + 8, specifies whether or not an interrupt signal should be sent to the GIC when the data present at an input port changes value. Setting a bit location in this register to 1 allows interrupts to be generated, while setting the bit to 0 prevents interrupts. Finally, the parallel port may contain an *Edgecapture* register at address *Base* + C. Each bit in this register has the value 1 if the corresponding bit location in the parallel port has changed its value from 0 to 1 since it was last read. Performing a write operation to the *Edgecapture* register sets all bits in the register to 0, and clears any associated interrupts.

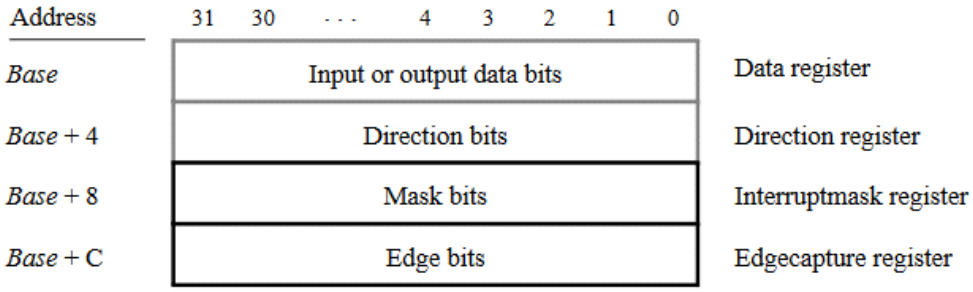

Figure 21. Registers used for interrupts from the parallel ports.

3.4.1 Interrupts from the Pushbutton Switches

Figure 12, reproduced as Figure 22, shows the registers associated with the pushbutton parallel port. The *Interruptmask* register allows interrupts to be generated when a key is pressed. Each bit in the *Edgecapture* register is set to 1 by the parallel port when the corresponding key is pressed. An interrupt service routine can read this register to determine which key has been pressed. Writing any value to the *Edgecapture* register deasserts the interrupt signal

mask register allows interrupts to be generated when a key is pressed. Each bit in the *Edgecapture* register is set to 1 by the parallel port when the corresponding key is pressed. An interrupt service routine can read this register to determine which key has been pressed. Writing any value to the *Edgecapture* register deasserts the interrupt signal being sent to the GIC and sets all bits of the *Edgecapture* register to zero.

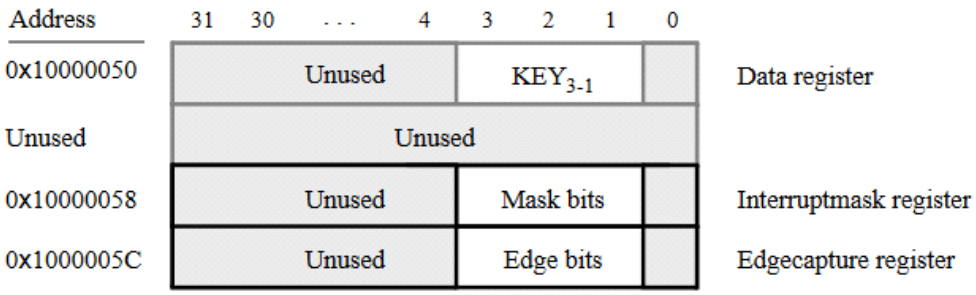

Figure 22. Registers used for interrupts from the pushbutton parallel port.