Quiz 1: Fifteen minutes

Design a 16-bit counter that either loads a new value or increments the current value by 4 at each rising clock edge. The counter accepts *clock*, a 16-bit *data-in* that is a 2's complement number, and *src* inputs. If *src* is 0, the counter stores the value on *data-in*; if *src* is 1, it increments the current value by 4.

Provide a schematic for the counter that uses high-level components (muxes, adders, registers) wherever possible. Label these components. You may use bus notation (as in the lab) to label connections between devices.

Quiz 1: Fifteen minutes

Several of this week's exercises asked you to create a *shifter*. You can combine a *shifter* and a *register* to create a *shift-register*: a device that stores a new value or shifts its current value.

Design a 16-bit shift-register that either loads a new value or rotates its current value at each rising clock edge. (Reminder: "Rotates" means that bits that are shifted "off the end" show up at the other end.) The shift-register accepts *clock*, a 16-bit *data-in,* and a 2-bit *op* inputs. If *op* is 00 or 01, the counter stores the value on *data-in*; if *src* is 10, it shifts the value one bit to the left; if *src* is 11, it shifts the value one bit to the right.

Provide a schematic for the shift-register that uses high-level components (muxes, adders, registers) wherever possible. Label these components. You may use bus notation (as in the lab) to label connections between devices. Note that you may hardwire a 0 or 1 if needed.

Quiz 1: Fifteen minutes

Design a 16-bit *accumulator* (a form of counter) that either loads a new value or accumulates a sum at each rising clock edge. The accumulator accepts *clock*, a 4-bit *data-in* that is a 2's complement value*,* and *src* inputs. If *src* is 0, the counter adds the current value to *data-in (*padding *data-in* appropriately to make it a sixteen bit value) and stores it; if *src* is 1, it pads *data-in* appropriately and stores it as the new value.

Provide a schematic for the *accumulator* that uses high-level components (muxes, adders, registers) wherever possible. Label these components. You may use bus notation (as in the lab) to label connections between devices. Note that you may hardwire a 0 or 1 if needed.

Quiz 1: Fifteen minutes

Design a 16-bit (2's complement) *up/down counter.* The counter accepts *clock*, *reset,* and *op-ctrl* inputs. If op-ctrl is 0, the counter adds 1 to its current value. If op-ctrl is 1, the counter subtracts 1 from its current value. If *reset* is received, the counter stores the value 0. *reset* takes priority over op-ctrl.

Provide a schematic for the *accumulator* that uses high-level components (muxes, adders, registers) wherever possible. *NOTE:* You may not use a register that has a *reset.* You must implement reset.

Label these components. You may use bus notation (as in the lab) to label connections between devices. Note that you may hardwire a 0 or 1 if needed.

You should strive to us as few hardware components as possible for this circuit. In particular: try to use a single adder, rather than two adders or an adder and subtracter, to implement the circuit.

Quiz 1: Fifteen minutes

Design a 16-bit (2's complement) *incrementer/doubler counter.* The counter accepts *clock*, *reset,* and *op-ctrl* inputs. If op-ctrl is 0, the counter adds 1 to its current value. If op-ctrl is 1, the counter doubles its current value. If *reset* is received, the counter stores the value 0. *reset* takes priority over op-ctrl.

Provide a schematic for the *accumulator* that uses high-level components (muxes, adders, registers) wherever possible. *NOTE:* You may not use a register that has a *reset.* You must implement reset. Also, you are restricted to a single adder for this circuit.

Label these components. You may use bus notation (as in the lab) to label connections between devices. Note that you may hardwire a 0 or 1 if needed.

Quiz 1: Fifteen minutes

Several of this week's exercises asked you to create a *shifter*. You can combine a *shifter* and a *register* to create a *shift-register*: a device that stores a new value or shifts its current value.

Design a 16-bit shift-register that either loads a new value or shifts its current value at each rising clock edge. The shift-register accepts *clock*, a 16-bit *data-in,* and a 2-bit *op* inputs. If *op* is 00, the counter stores the value on *data-in*; if *src* is 01, it shifts the value one bit to the left; if *src* is 10 or 11, it shifts the value one bit to the right. All shifts should be *logical shifts.*

Provide a schematic for the shift-register that uses high-level components (muxes, adders, registers) wherever possible. Label these components. You may use bus notation (as in the lab) to label connections between devices. Note that you may hardwire a 0 or 1 if needed.