The Integrator

William E. Earle

August 31, 2009

1 General

The integrator part of the STM Z-box logic is a 32 bit up-down counter driven by a 12.5 MHZ clock and controlled by four input signals, UP, DOWN, RESET and HOLD. When the RESET input is high on the rising edge of the clock, the counter is set to its zero value (MSB high and all other bits low), irrespective of the state of the other control bits. If the HOLD input is high and RESET is low, then the count does not change irrespective of the UP or DOWN inputs. If both RESET and HOLD are low, and if the UP or DOWN input to the counter is high on the rising edge of the clock, then the counter is incremented or decremented respectively. The counter has a maximum value of all 1’s (maximum positive integral) and a minimum value of all zeroes (maximum negative integral) and will neither overflow nor underflow.

A four bit length setting (SEL(0) through SEL(3)) is also input by the counter to reduce the counter length from 32 down to as few as 17 bits. With these four bits low, the counter has a length of 32 (minimum integrator gain), and with the bits high the length is 17 (maximum integrator gain).

Only the upper 20 bits of the counter are used for output to the DAC or to the controlling PC. Thus, if the counter is set to a length of 17, the lower three bits output to the DAC or PC remain unchanged.
2 Implementation Approach

One of the difficulties with this counter design is the need for a variable length from 32 to 17 bits. This is not a problem for short counters, but for longer ones, such as 32 bits, the logic gets quite complex and may not be synthesizable in a reasonable size CPLD. To avoid this complexity, the counter is not implemented using conventional counter logic but rather by using a register with feedback through an adder to provide the increment and decrement functions.

The decremerter can be effected in several ways. One way is to use the same adder and change the addend to the 2’s complement of the incrementer bit. Another approach is to use a separate adder and again make the addend the 2’s complement of the incrementer bit; in this approach both the increment and decrement of the present count in the register are simultaneously available and the selection of the particular adder output determines whether the value in the register (the counter) is incremented or decremented. A third and similar approach is to actually implement a subtractor and again choose the output of the adder or subtractor as feedback to the register. However, since subtractors are less familiar than adders (but no more complicated), either the first or second approach is preferable.

The actual implementation uses the second approach with two adders, partly because of logic simplification but mostly because both the increment and decrement values are simultaneously available for feedback to the register, hence there is no additional delay when switching between incrementing and decrementing. Using two adders may seem like a circuit complication, but the adders are identical so no additional design time is required.

3 Generating the Addend for the Subtractor

For the incremerter, the addend is simply a one to the adder bit selected by the four bits that specify the counter length. For the subtractor, the addend is the 2’s complement of this same bit. A very simple way to generate this value is to set the addend to all ones except for the selected bit and to then add a carry into the LSB of the adder.
4 Speeding Up the Adder

A 32 bit ripple carry adder will not be able to generate a sum in the 80ns minimum time between counts, so the adder is implemented using three-level carry-look-ahead logic.