Multi-function I/O

Features

16-bit up/down counter

Count limit (for up/down rollover) can be any value

Four external bidirectional connections

As outputs, can be used for PWM, PPM, and synchronized output

As inputs, can be used for quadrature decode, width measurement, and counting

Standard byte-wide interface

Description

The core of each channel is a sixteen-bit counter that can be clocked either by the master clock for the device or an external prescaler.

This counter is accompanied by a number of registers that are updated with control register values each time the counter rolls over. This buffering allows the control registers to be updated with values to be used during the next count cycle during the current count cycle. The registers that are buffered in this way are the sixteen-bit reload register and the four sixteen-bit match registers.

The four sixteen-bit match registers each generate an output pulse when the count is equal to the value programmmed in the match registers. These pulses can be used to set or reset any of the outputs from the channel.

The sixteen-bit limit register determines when the counter will be reloaded. When incrementing, if the next count will be greater than the contents of the limit register the counter is cleared. In those cases where the counter is decrementing the counter is reloaded when the next count will be less than zero. The counter is reloaded with the limit register value in this case.

The channel can also be used as a simple input or output port. In this case the timer can still be used to generate periodic interrupts separately.

Interface

```
module chan_top (chxx_int, chxx_pen, chxx_pout, chxx_rbus, chxx_a0, chxx_cmd, chxx_pin,
                 chxx_sel, clk, cntr_en, glob_sync, iwrite_bus, peri_rd, peri_wr, prot_clk,
                 resetb);

  input         chxx_a0;       /* channel xx address                                    */
  input         chxx_sel;      /* channel xx enable                                     */
  input         clk;           /* main clock                                            */
  input         cntr_en;       /* count enable                                          */
  input         glob_sync;     /* global sync                                           */
  input         peri_rd;       /* peripheral read strobe                                */
  input         peri_wr;       /* peripheral write strobe                               */
  input         prot_clk;      /* protection clock                                      */
  input         resetb;        /* internal reset                                        */
  input   [1:0] chxx_cmd;      /* channel xx commands                                   */
  input   [3:0] chxx_pin;      /* channel xx port data in                               */
  input   [7:0] iwrite_bus;    /* internal write data bus                               */
  output        chxx_int;      /* channel xx interrupt request                          */
  output  [3:0] chxx_pen;      /* channel xx port enable                                */
  output  [3:0] chxx_pout;     /* channel xx port out                                   */
  output  [7:0] chxx_rbus;     /* channel xx read data bus                              */
```