This document describes a NIM module which is designed to facilitate synchronization of a modulated IR laser with a 10Hz YAG Laser for use in laser absorption studies. The description applies in particular to a Rev 2 design completed in October, 2004 with a printed circuit board (vs the hand-wired Rev 1 unit).

Essentially the unit accepts a 1kHz reference pulse train (trigger) input and provides flashlamp, Q-switch and shutter outputs derived using dividers and delays. All outputs are positive, TTL-compatible 5V signals unless otherwise noted.

**Trigger Input** (isolated BNC, front panel)
This is an input designed to accept positive pulses in the 0-5V range. The input impedance is 2k ohms. The ground shield connector on the BNC is isolated, and the input is differential, so any possible ground loops are broken and common-mode noise rejected.

**Trig Ref Out** (BNC, rear panel)
This is a 5V TTL-compatible inverted copy of the trigger input. It's primary use is for viewing on an oscilloscope to ensure that the trigger signal is being received cleanly.

**Trig Delay Out** (BNC, front panel)
This is a delayed version of the trigger input. The delay is adjustable from roughly 0.4 to >1 ms. (setting the delay beyond 1ms causes the output to disappear because of the way the delay device works). The pulse width is fixed at about $28 \mu s$.

**Flashlamp Out** (BNC, rear panel and also DB-9 pin 4)
This is the delayed trigger (Trig Delay Out) divided by 100, such that for a 1kHz trigger input the output pulses at 10Hz. The pulse width is fixed at about $28 \mu s$. The output may be disabled by means of a front-panel switch.

**Q-Switch Out** (BNC, front panel)
This is similar to the Flashlamp Output, with an additional delay adjustable from 80$\mu$s - 1.5ms from the delayed trigger (divided by 100).

**Shutter Out** (BNC, front panel)
This is a signal derived from the 10Hz divided trigger, delayed, and optionally gated by the Linewidth Input. The width is fixed at about 77ms. The Shutter Out may be set to occur at a rate of 1/2 Hz, 1Hz or 2Hz by means of a 3-position front-panel switch. An additional divide-by-10 may be enabled by means of another front panel switch giving a pulse every 5, 10 or 20 seconds. The divider output is subject to an adjustable delay of 100ms - 200ms.

When the Normal/Linewidth switch is in the Linewidth position, the Shutter Out is gated by an external signal on the Linewidth Input. In this switch position, a low (0V) level on the Linewidth Input will enable the Shutter Out.
Technical Description

The Trigger Input is received differentially using an OPA651 wideband op-amp. The input BNC ground is isolated from circuit ground. The gain is +1 on the center conductor and -1 on the ground. Since this is a rather high-bandwidth device any ringing on the input may cause multiple transitions to appear on the Trig Ref Output. This should not affect operation of the remainder of the circuit, as the delay one-shot is edge-triggered. The op-amp output is buffered by a 74AC04 inverter, driving the Trig Ref Output, and one input of the CPLD.

All delays and output pulse shaping are performed with 74AHCT123 dual one-shots. These have the advantage over various similar devices that the output pulse width is exactly equal to R * C (modulo some temperature effects). Thus, the delay and pulse width values may be changed in a straightforward way if necessary by changing the appropriate timing capacitors and resistors.

All logic except for the delays is contained on an Altera MAX3064 CPLD device. This is a 64-macrocell programmable logic device with integral flash memory. It may be reprogrammed if necessary by connecting an Altera Byte Blaster MV programming cable to the 10-pin connector on the rear panel and using appropriate software from Altera.

The logic inside the MAX3064 was implemented using Quartus II v4.1 Web Edition software which is freely available from the Altera website. The design consists of a top-level schematic, and four VHDL design files. Two of the VHDL files (div100.vhd and div_flash.vhd) were custom-written, and their listings are included in this document. The other two files (oscdiv.vhd and lpm_mux0.vhd) were generated by the Altera megafunction wizard).
Illustration ICPLD Logic Design Schematic
VHDL Code Listing for div100 block

-- This code implements a divide-by-100 counter with an output
-- pulse 5 clock cycles wide

LIBRARY ieee;
USE ieee.std_logic_1164.all;

-- Entity Declaration
ENTITY div100 IS
-- {{ALTERA_IO_BEGIN}} DO NOT REMOVE THIS LINE!
PORT
  (  
    clkin : IN STD_LOGIC;
    grst_n : IN STD_LOGIC;
    divout : OUT STD_LOGIC  
  );
-- {{ALTERA_IO_END}} DO NOT REMOVE THIS LINE!
END div100;

-- Architecture Body
ARCHITECTURE div100_architecture OF div100 IS

  signal cnt : integer range 0 to 99;
  signal divout_s : std_logic;

BEGIN

  process (clkin, grst_n)
  begin  -- process
    if grst_n = '0' then                -- asynchronous reset (active low)
      cnt <= 0;
      divout_s <= '0';
    elsif clkin'event and clkin = '1' then  -- rising clock edge
      cnt <= cnt + 1;
      if cnt = 99 then
        cnt <= 0;
        divout_s <= '0';
      end if;
      if cnt = 4 then
        divout_s <= '1';
      end if;
    end if;
  end process;

  divout <= divout_s;

END div100_architecture;
VHDL Code Listing for div_flash - Flash Lamp Divider

-- divider for flash lamp output
-- gets 10Hz input from main clock divider
-- controlled by two switches:
--   three-position SPDT switch with two switch-to-ground contacts
--     sw_up  - 1/2 Hz when '0'
--     sw_dn  - 2 Hz   when '0'
--     (none) - 1 Hz   (both '1')
--   second switch enables an additional divide-by-10 when '0'
--
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

-- Entity Declaration
entity div_flash is
  -- {{ALTERA_IO_BEGIN}} DO NOT REMOVE THIS LINE!
  port
  (    clkin  : in  std_logic;
    grst_n : in  std_logic;
    sw_up  : in  std_logic;
    sw_dn  : in  std_logic;
    sw_10  : in  std_logic;
    divout : out std_logic
  );
  -- {{ALTERA_IO_END}} DO NOT REMOVE THIS LINE!
end div_flash;

-- Architecture Body
architecture div_flash_architecture of div_flash is

signal div5            : integer range 0 to 4;
signal div4            : std_logic_vector(1 downto 0);
signal div10           : integer range 0 to 9;
signal div5_s, div10_s : std_logic;
signal div_a, div_a0   : std_logic;

begin

process (clkin, grst_n)
begin  -- process
  if grst_n = '0' then                    -- asynchronous reset (active low)
    div5     <= 0;
    div10    <= 0;
    div5_s   <= '0';
    div10_s  <= '0';
    div4     <= "00";
  elsif clkin'event and clkin = '1' then  -- rising clock edge
    -- divide by 5 (2Hz out on div5_s)
    div5     <= div5 + 1;
    if div5 = 2 then
      div5_s <= '1';
    end if;
  end if;
end process;

begin
  -- process
  if grst_n = '0' then                    -- asynchronous reset (active low)
    div5     <= 0;
    div10    <= 0;
    div5_s   <= '0';
    div10_s  <= '0';
    div4     <= "00";
  elsif clkin'event and clkin = '1' then  -- rising clock edge
    -- divide by 5 (2Hz out on div5_s)
    div5     <= div5 + 1;
    if div5 = 2 then
      div5_s <= '1';
    end if;
end process;

end div_flash_architecture;
if div5 = 4 then
  div5_s <= '0';
  div5   <= 0;
  -- divide by 4 (1Hz, 1/2 Hz on div4(0), div4(1)
  div4   <= div4 + 1;
end if;

div_a0 <= div_a;

-- multiplex 1/2, 1, 2Hz on div4_a
if sw_up = '0' then
  div_a <= div4(1);               -- 1/2 Hz
elsif sw_dn = '0' then
  div_a <= div5_s;                -- 2 Hz
else
  div_a <= div4(0);               -- 1 Hz
end if;

-- divide-by-10
if div_a0 = '0' and div_a = '1' then
  div10  <= div10 + 1;
  if div10 = 4 then
    div10_s <= '1';
  end if;
  if div10 = 9 then
    div10 <= 0;
    div10_s <= '0';
  end if;
end if;

-- multiplex output
if sw_10 = '0' then
  divout <= div10_s;
else
  divout <= div_a;
end if;

end if;
end process;
end div_flash_architecture;