

Mueller, Carsten

Von: Geissler, Rene
Gesendet: Mittwoch, 8. Mai 2024 12:25
An: Mueller, Carsten
Betreff: AW: Konstanten..

Hallo Carsten,

ich gebe zu, dass die Benennung der Variablen nicht sehr aussagekräftig ist;)

>> Die Laufvariable „r“ wird aus den ersten drei Stellen des Bitstrings zu einer dreistelligen Integerzahl abgeleitet für Binärkodierte Anzeige.

Nein, die Variable r wird nur für den "Balken"-Modus 0 verwendet, um die entsprechende Anzahl an LEDs leuchten zu lassen. Dass die Integerzahl dreistellig ist, stimmt, ist aber hierfür nicht von Belang.

>>Mode1 greift die drei Rangebits wie sie rein kommen, eben binär codiert ab 1 und eben nur die ersten drei. Test und Kanal Präsent werden separat behandelt (s = 5?)

Ja.

VG
René

Von: Mueller, Carsten
Gesendet: Mittwoch, 8. Mai 2024 11:48:34
An: Geissler, Rene
Betreff: AW: Konstanten..

Hallo Rene,

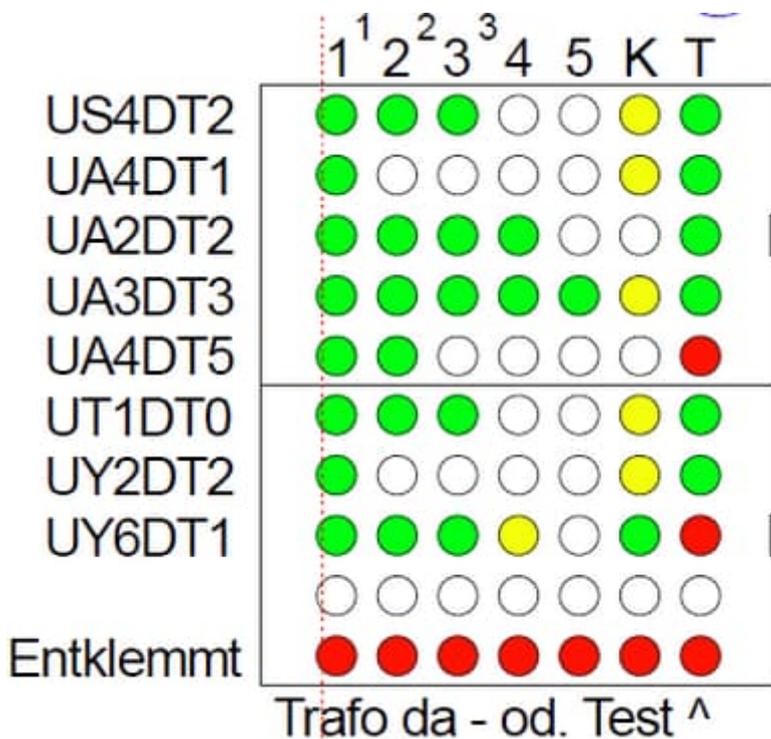
ich bin da noch nicht ganz sicher was den CPLD1 angeht:

Der 1ms-Takt wird gezählt vom CPLD2 und zählt von 0..19 als Laufvariable"r".

Die Daten für die acht Trafos 0..7 kommen von außen über die Pin-Zuordnung über das Package als Bitzeichenfolge wie „10001“ (Rb0 Rb1 Rb2 Rb3=Test Rb4=Präsent oder kurz „RRRT“).

Dieser String ist zunächst einmal keine Repräsentation einer Integerzahl (nur in den ersten drei Stellen)!

Das ist aber nicht die Reihenfolge der Anzeige im Display, denn die differiert je nach Mode und es gibt den Klemmpuls noch. Deswegen unterscheidest Du Mode 0 und Mode 1:



Der LED-Treiber-Prozess soll nun die richtige Zuordnung auf die LEDs mit dem Stroke-Takt leisten.

Die Laufvariable „m“ läuft durch die Anzahl der fünf Messbereiche 0..4.

Die Laufvariable „r“ wird aus den ersten drei Stellen des Bitstrings zu einer dreistelligen Integerzahl abgeleitet für Binärkodierte Anzeige.

Wie resultiert daraus ein Bitbalken mit bis zu 5 LEDs?

Die Binär-Repräsentation der MBs wird aber auch von 1..5 gezählt!

Mode0 bearbeitet in seiner Schleife wohl den Bitstring wie er rein kommt, um das jeweilige Bit hinter „s“ im 1ms-Takt in der richtigen Spalte zu aktivieren.

Also bei Zeit s:

s = 0: Rb0

s = 1: Rb1

s = 2: Rb2

s = 6: Test=rot

s = 6: Trafo-Präsent=grün

Wie entsteht der Bitbalken? m durchläuft die fünf möglichen Anzeigestellen für den Range und setzt solange 1sen bis der Index <= dem höchsten ist?

Mode1 greift die drei Rangebits wie sie rein kommen, eben binär codiert ab 1 und eben nur die ersten drei. Test und Kanal Präsent werden separat behandelt (s = 5?)

Ist damit also „<=“ richtig und „>=“ falsch?

Hab ich das sonst richtig verstanden?

Schönen Gruß,

Carsten

Von: Geissler, Rene <R.Geissler@gsi.de>

Gesendet: Mittwoch, 8. Mai 2024 09:52

An: Mueller, Carsten <Ca.Mueller@gsi.de>
Betreff: AW: Konstanten..

Hallo Carsten,

ich hatte gestern kurz nach der ersten - leider fehlerhaften - Korrektur noch eine zweite Korrektur im Git-Repo vorgenommen:

https://git.gsi.de/BEA_HDL/MAPS_CPLD_Gateway/-/blob/master/src/vhdl/cpld_1.vhd?ref_type=heads

An der ursprünglichen Version war denke ich nur das -1 falsch. Als statt dem Stand von vorgestern:

```
if m < c_measurement_range_wl - 1 then
```

ist denke ich:

```
if m < c_measurement_range_wl then
```

richtig, so dass das dritte Bit auch dargestellt wird. Oder ich hätte ein kleiner/gleich verwenden müssen.

VG
René

Von: Mueller, Carsten
Gesendet: Mittwoch, 8. Mai 2024 08:57:04
An: Geissler, Rene
Betreff: Konstanten..

Guten Morgen Rene,

Eine Frage zu zwei Konstanten:

```
package constant_package is

    constant c_num_trafos          : integer := 8;
    constant c_trafo_signal_wl    : integer := 5; -- z.B. "RRRTP" (K) (im
    constant c_measurement_range_wl : integer := 3; -- 5? z.B. "00111.1" + P
    constant c_manual_range_wl    : integer := 4;
    constant c_num_measurement_ranges : integer := 5; -- 3? z.B. &d5 = &b"101"
    constant c_num_1ms_strokes    : integer := 20;
    constant c_num_20ms_periods   : integer := 50;
```

CPLD1 Codeteil:

(Klemmpulse in Spalte 6 werden von diesem CPLD1 nicht abgehandelt)

```
for t in 0 to c_num_trafos - 1 loop
  -- common to modes 0 and 1
  if mode <= to_unsigned(1, c_mode_wl) then
    r := to_integer(unsigned(trafo_signals(t)(c_measurement
    for m in 0 to c_num_measurement_ranges - 1 loop
      if s = m or s = 6 + m then
        -- mode 0
        if mode = to_unsigned(0, c_mode_wl) then
          if r >= m + 1 then
            led_rows_green(t) <= '1';
          end if;
        -- mode 1
        else
          if m < c_num_measurement_ranges
            led_rows_green(t) <= tr
          end if;
        end if;
      end if;
    end loop;
    if s = 5 or s = 6 + 5 then
      led_rows_green(t) <= trafo_signals(t)(4); -- d
      --led_rows_red(t) <= trafo_signals(t)(3); (Testf
    end if;
  end if;
end loop;
```

Müssten die Integerwerte der Gelb umrahmten Konstanten nicht genau umgekehrt sein wie im grauen?

Schönen Gruß,

Carsten