axis_1553_encoder.v

AUTHORS

JAY CONVERTINO

DATES

2021/05/17

INFORMATION

Brief

AXIS MIL-STD-1553 ENCODER

License MIT

axis_1553_encoder

```
module axis_1553_encoder #(
parameter
CLOCK_SPEED
 =
2000000,
parameter
SAMPLE_RATE
 =
2000000
) ( input aclk, input arstn, input [15:0] s_axis_tdata, input s_axis_tvalid,
```

AXI streaming to MIL-STD-1553 encoder. This encoder can be used at 2 Mhz or above. TDATA is 16 bit data to be transmitted. TUSER sets how the core works.

Parameters

CLOCK_SPEED parameter — This is the aclk frequency in Hz, must be 2 MHz or above.

SAMPLE_RATE parameter — 2 MHz or above rate that is an even divisor of CLOCK_SPEED

Ports

aclk — Clock for all logic

arstn — Negative reset

s_axis_tdata — Input data for 1553 encoder.

s_axis_tvalid — When set active high the input data is valid.

s_axis_tuser — Information about the AXIS data {TYY,NA,I,P}

Bits explained below:

```
- TYY = TYPE OF DATA
      - 000 = NA
      - 001 = REG (NOT IMPLIMENTED)
      - 010 = DATA
      - 100 = CMD/STATUS
- D   = DELAY ENABLED
- I   = INVERT DATA
- P   = PARITY
      - 1 = ODD
      - 0 = EVEN
```

s_axis_tready — When active high the device is ready for input data.

diff — Output data in TTL differential format.

en_diff — When diff is valid data, this is active high and can be used to switch a mux.

base_1553_clock_rate

```
localparam integer base_1553_clock_rate = 1000000
```

1553 base clock rate

samples_per_mhz

```
localparam integer samples_per_mhz = SAMPLE_RATE / base_1553_clock_rate
```

sample rate to caputre transmission bits at

cycles_per_mhz

```
localparam integer cycles_per_mhz = CLOCK_SPEED / base_1553_clock_rate
```

calculate the number of cycles the clock changes per period

samples_to_skip

```
localparam integer samples_to_skip = (
 (cycles_per_mhz > samples_per_mhz) ? cycles_per_mhz / samples_per_mhz -
1
:
0
)
```

calculate the number of samples to skip

bit_rate_per_mhz

```
localparam integer bit_rate_per_mhz = samples_per_mhz
```

bit rate per mhz

delay_time

```
localparam integer delay_time = cycles_per_mhz * 4
```

delay time, 4 is for 4 us (min 1553 time)

sync_pulse_len

```
localparam integer sync_pulse_len = bit_rate_per_mhz * 3
```

sync pulse length

bits_per_trans

```
localparam integer bits_per_trans = 20
```

bits per transmission

synth_bits_per_trans

```
localparam integer synth_bits_per_trans = (
bits_per_trans*bit_rate_per_mhz
)
```

sync bits per trans

bit_pattern

```
localparam [(
bit_rate_per_mhz
)-1:0]bit_pattern = {{bit_rate_per_mhz/2{1'b1}}, {bit_rate_per_mhz/2{1'b0}}
```

create the bit pattern. This is based on outputing data on the negative and positive. This allows the encoder to run down to 1 mhz.

synth_clk

```
localparam [synth_bits_per_trans-1:0]synth_clk = {
bits_per_trans{bit_pattern}
 }
```

synth clock is the clock constructed by the repeating the bit pattern. this is intended to be a representation of the clock. Captured at a bit_rate_per_mhz of a 1mhz clock.

sync_cmd_stat

```
localparam [sync_pulse_len-1:0]sync_cmd_stat = {
                                                                         {
sync_pulse_len/2{1'b0}},
                                                                         {
sync_pulse_len/2{1'b1}}
}
```

sync pulse command

sync_data

```
localparam [sync_pulse_len-1:0]sync_data = {
                                                                         {
sync_pulse_len/2{1'b1}},
                                                                         {
sync_pulse_len/2{1'b0}}
}
```

sync pulse data

cmd_data

```
localparam cmd_data = 3'b010
```

tuser decode for data

cmd_data

tuser decode for command

cmd_data

enable diff output

STATE MACHINE

Variables that makeup the encoder state machine.

data_cap

```
localparam data_cap = 3'd1
```

data capture

data_invert

```
localparam data_invert = 3'd2
```

invert data

parity_gen

```
localparam parity_gen = 3'd3
```

parity generator

process

```
localparam process = 3'd4
```

command processor

pause_ck

```
localparam pause_ck = 3'd5
```

check for pause (4us)

trans

```
localparam trans = 3'd6
```

transmit data

error

```
localparam error = 3'd0
```

someone made a whoops