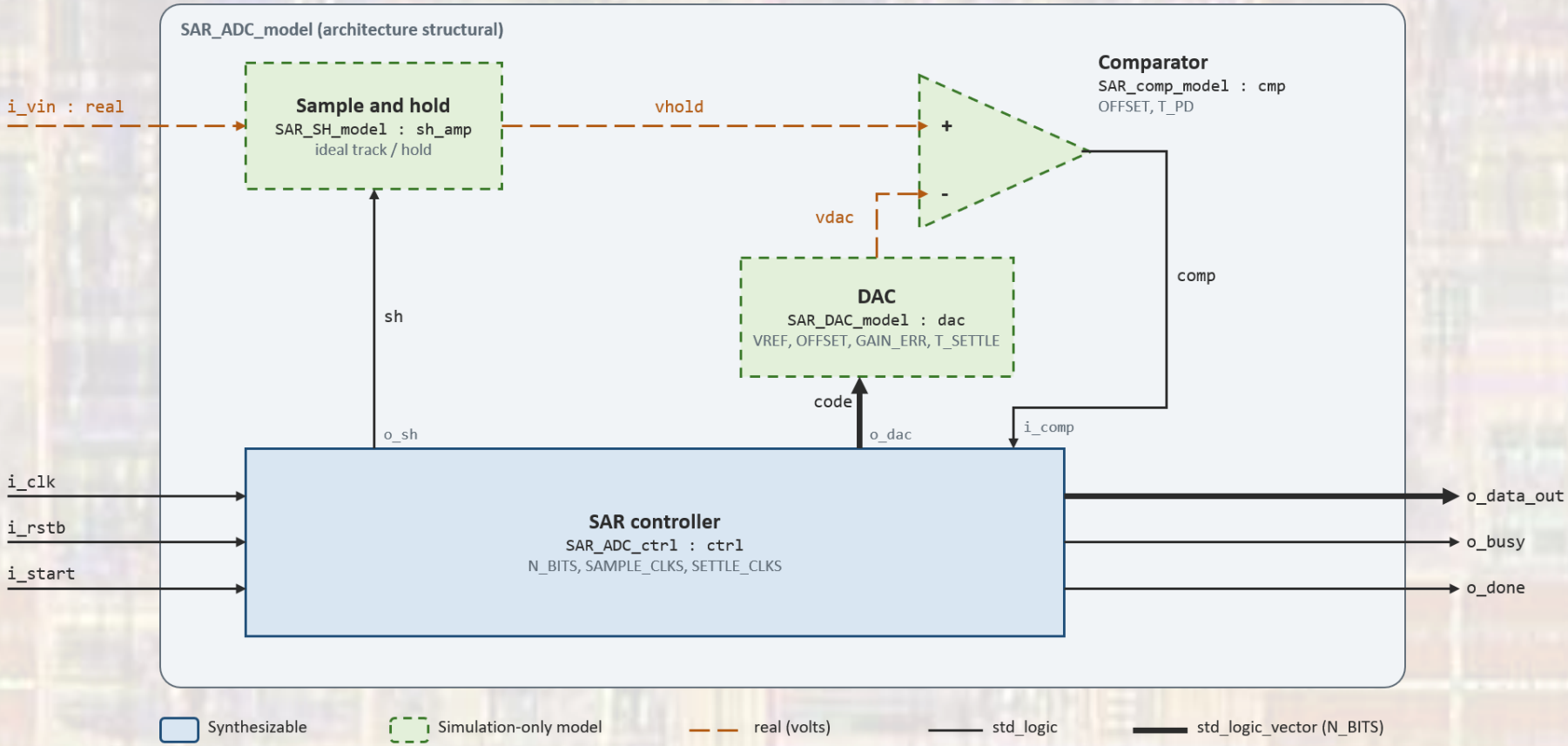
ADC – SAR Model

Last updated 10/8/26

ADC – SAR Model

- SAR Type ADC Model
  - Model because it contains “real” signals that can’t be synthesized



 © tj

ADC – SAR Model

- SAR Type ADC Model – S/H Description

```
----------------------------------------
--
-- SAR_SH_model.vhdl
--
-- created 10/6/26
-- tj
--
-- rev 0
----------------------------------------
--
-- Behavioral sample and hold model for SAR ADC simulation
--
-- sh = '1' -> output tracks the input
-- sh = '0' -> output holds the last value
--
----------------------------------------
--
-- Inputs:  vin (real - volts), sh
-- Outputs: vhold (real - volts)
--
-- NOTE: SIMULATION ONLY - not synthesizable (real ports)
-- NOTE: ideal - no droop, acquisition time, or pedestal error
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;
```


© tj

ADC – SAR Model

- SAR Type ADC Model – S/H VHDL
  - Implemented as an analog register

```
entity SAR_SH_model is
    port(
          i_vin:      in  real;
          i_sh:       in  std_logic;

          o_vhold:    out real
        );
end entity;

architecture mixed_signal of SAR_SH_model is
    -- this is effectively an analog latch
    -- no latch inferred because we are not sythesizing it
begin
    process(i_vin, i_sh)
    begin
        if(i_sh = '1') then
            o_vhold <= i_vin;
        end if;
    end process;

end architecture;
```


© tj

ADC – SAR Model

- SAR Type ADC Model – Comparator Description

```
----------------------------------------
--
-- SAR_comp_model.vhdl
--
-- created 10/6/26
-- tj
--
-- rev 0
----------------------------------------
--
-- Behavioral comparator model for SAR ADC simulation
--
-- comp = '1' when vp >= (vn + OFFSET), else '0'
-- output changes T_PD after the inputs change
--
----------------------------------------
--
-- Inputs:  vp, vn (real - volts)
-- Outputs: comp
--
-- NOTE: SIMULATION ONLY - not synthesizable (real ports)
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;
```


© tj

ADC – SAR Model

- SAR Type ADC Model – Comparator VHDL
  - Includes propagation delay and input offset error

```
entity SAR_comp_model is
    generic(
          OFFSET:  real := 0.0;
          T_PD:    time := 3 ns
          );
    port(
          i_vp:    in  real;
          i_vn:    in  real;
          o_comp:  out std_logic
       );
end entity;

architecture mixed_signal of SAR_comp_model is
begin
    process(i_vp, i_vn)
    begin
       if(i_vp >= (i_vn + OFFSET)) then
          o_comp <= '1' after T_PD;
       else
          o_comp <= '0' after T_PD;
       end if;
    end process;

end architecture;
```


© tj

ADC – SAR Model

- SAR Type ADC Model – DAC Description

```
----------------------------------------
--
-- SAR_DAC_model.vhdl
--
-- created 10/6/26
-- tj
--
-- rev 0
----------------------------------------
--
-- Behavioral DAC model for SAR ADC simulation
--
-- Vdac = (VREF * code / 2^N) * (1 + GAIN_ERR) + OFFSET
-- output updates T_SETTLE after the code changes
--
----------------------------------------
--
-- Inputs:  code
-- Outputs: vdac (real - volts)
--
-- NOTE: SIMULATION ONLY - not synthesizable (real ports)
-- NOTE: T_SETTLE is the DAC delay
--       code changes during the delay are ignored
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
```


© tj

ADC – SAR Model

- SAR Type ADC Model – DAC VHDL
  - Includes gain and offset error and settling time

```
entity SAR_DAC_model is
   generic(
         N_BITS:     positive := 8;
         VREF:       real     := 3.3;
         OFFSET:     real     := 0.0;
         GAIN_ERR:   real     := 0.0;        -- fraction, 0.01 = 1%
         T_SETTLE:   time     := 5 ns
         );
   port(
         i_code:     in  std_logic_vector((N_BITS - 1) downto 0);
         o_vdac:     out real
      );
end entity;

architecture mixed_signal of SAR_DAC_model is
begin
   process(i_code)
   -- variable allowed due to this being a model - not synthesized
   -- reminder - variables update instantly in processes
   variable v: real;
   begin
      -- detect if a signal is set to 'unknown' fn: is_x(signal)
      if(is_x(i_code)) then
         v := 0.0;          -- default output if i_code is undefined
      else
         v := VREF * real(to_integer(unsigned(i_code))) / real(2**N_BITS);
         v := v * (1.0 + GAIN_ERR) + OFFSET;
      end if;

      o_vdac <= v after T_SETTLE;
   end process;

end architecture;
```


© tj

ADC – SAR Model

- SAR Type ADC Model – Controller
  - See the Controller slides

© tj

ADC – SAR Model

- SAR Type ADC Model – Description

```
----------------------------------------
--
-- SAR_ADC_model.vhdl
--
-- created 10/6/26
-- tj
--
-- rev 0
----------------------------------------
--
-- Complete SAR ADC model for controller testing
--
-- Combines:
--    SAR_ADC_ctrl   - synthesizable controller
--    SAR_SH_model   - behavioral sample and hold
--    SAR_DAC_model  - behavioral DAC
--    SAR_comp_model - behavioral comparator
--
--               +------+    vhold    +------+
--   i_vin ----->| S/H  |------------>|+     |
--               +------+             | comp |---+
--                  ^          +----->|-     |   |
--                  | sh       | vdac +------+   |
--               +------+   +-----+               |
--               | ctrl |-->| DAC |               |
--               +------+   +-----+               |
--                  ^   code                      |
--                  +-----------------------------+
--
----------------------------------------
--
-- Inputs:  clk, rstb, start, vin (real - volts)
-- Outputs: data_out, busy, done
--
-- NOTE: SIMULATION ONLY - not synthesizable (real ports)
-- NOTE: T_SETTLE + T_PD must be less than SETTLE_CLKS clock periods
--       in this case Tclk = 20ns
-- NOTE: Using the VHDL 2008 shortcut for component/instantiation
--
----------------------------------------
```


© tj

ADC – SAR Model

- SAR Type ADC Model – Entity

```
library ieee;
use ieee.std_logic_1164.all;

entity SAR_ADC_model is
    generic(
            N_BITS:         positive := 8;
            SAMPLE_CLKS:    positive := 4;
            SETTLE_CLKS:    positive := 1;
            VREF:           real := 3.3;
            DAC_OFFSET:     real := 0.0;
            DAC_GAIN_ERR:   real := 0.0;
            DAC_T_SETTLE:   time := 5 ns;
            COMP_OFFSET:    real := 0.0;
            COMP_T_PD:      time := 3 ns
        );
    port(
          i_clk:       in  std_logic;
          i_rstb:      in  std_logic;
          i_start:     in  std_logic;
          i_vin:       in  real;

          o_data_out: out std_logic_vector((N_BITS - 1) downto 0);
          o_busy:      out std_logic;
          o_done:      out std_logic
      );
end entity SAR_ADC_model;
```


© tj

ADC – SAR Model

- SAR Type ADC Model – Structural VHDL

```
architecture structural of SAR_ADC_model is

   -- internal "analog" and control signals
   signal sh:     std_logic;
   signal comp:   std_logic;
   signal code:   std_logic_vector((N_BITS - 1) downto 0);
   signal vhold:  real := 0.0;
   signal vdac:   real := 0.0;

begin

   ctrl: entity work.SAR_ADC_ctrl
      generic map(
            N_BITS      => N_BITS,
            SAMPLE_CLKS => SAMPLE_CLKS,
            SETTLE_CLKS => SETTLE_CLKS
            )
      port map(
            i_clk          => i_clk,
            i_rstb         => i_rstb,
            i_start        => i_start,
            i_comp         => comp,
            o_sh           => sh,
            o_sar_to_dac   => code,
            o_data_out     => o_data_out,
            o_busy         => o_busy,
            o_done         => o_done
         );

   sh_amp: entity work.SAR_SH_model
      port map(
            i_vin    => i_vin,
            i_sh     => sh,
            o_vhold  => vhold
         );
```

```
   dac: entity work.SAR_DAC_model
      generic map(
            N_BITS    => N_BITS,
            VREF      => VREF,
            OFFSET    => DAC_OFFSET,
            GAIN_ERR  => DAC_GAIN_ERR,
            T_SETTLE  => DAC_T_SETTLE
            )
      port map(
            i_code    => code,
            o_vdac    => vdac
         );

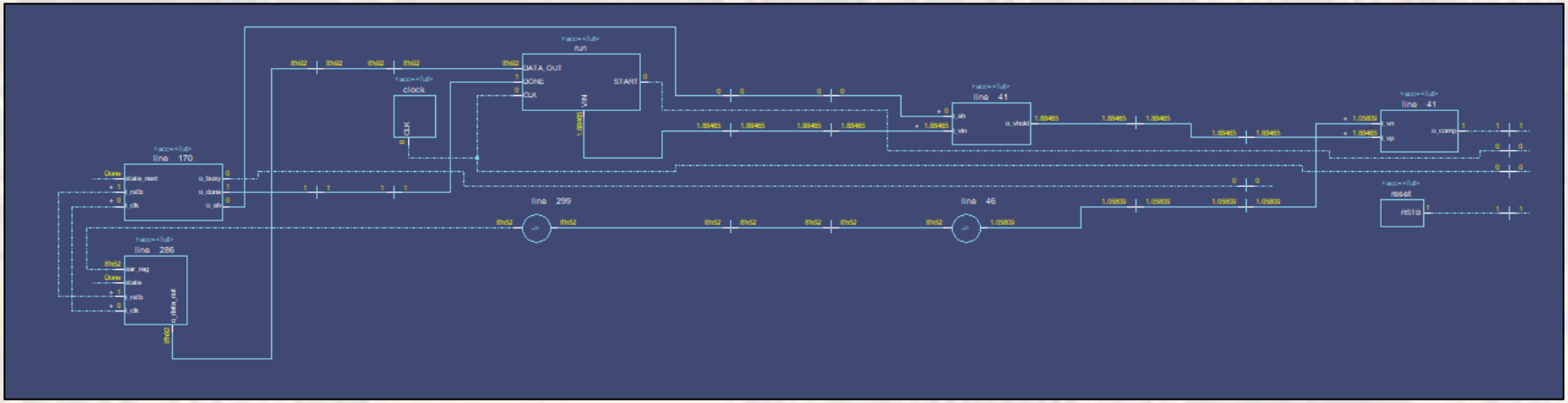
   cmp: entity work.SAR_comp_model
      generic map(
            OFFSET    => COMP_OFFSET,
            T_PD      => COMP_T_PD
            )
      port map(
            i_vp      => vhold,
            i_vn      => vdac,
            o_comp    => comp
         );

end architecture;
```


© tj

ADC – SAR Model

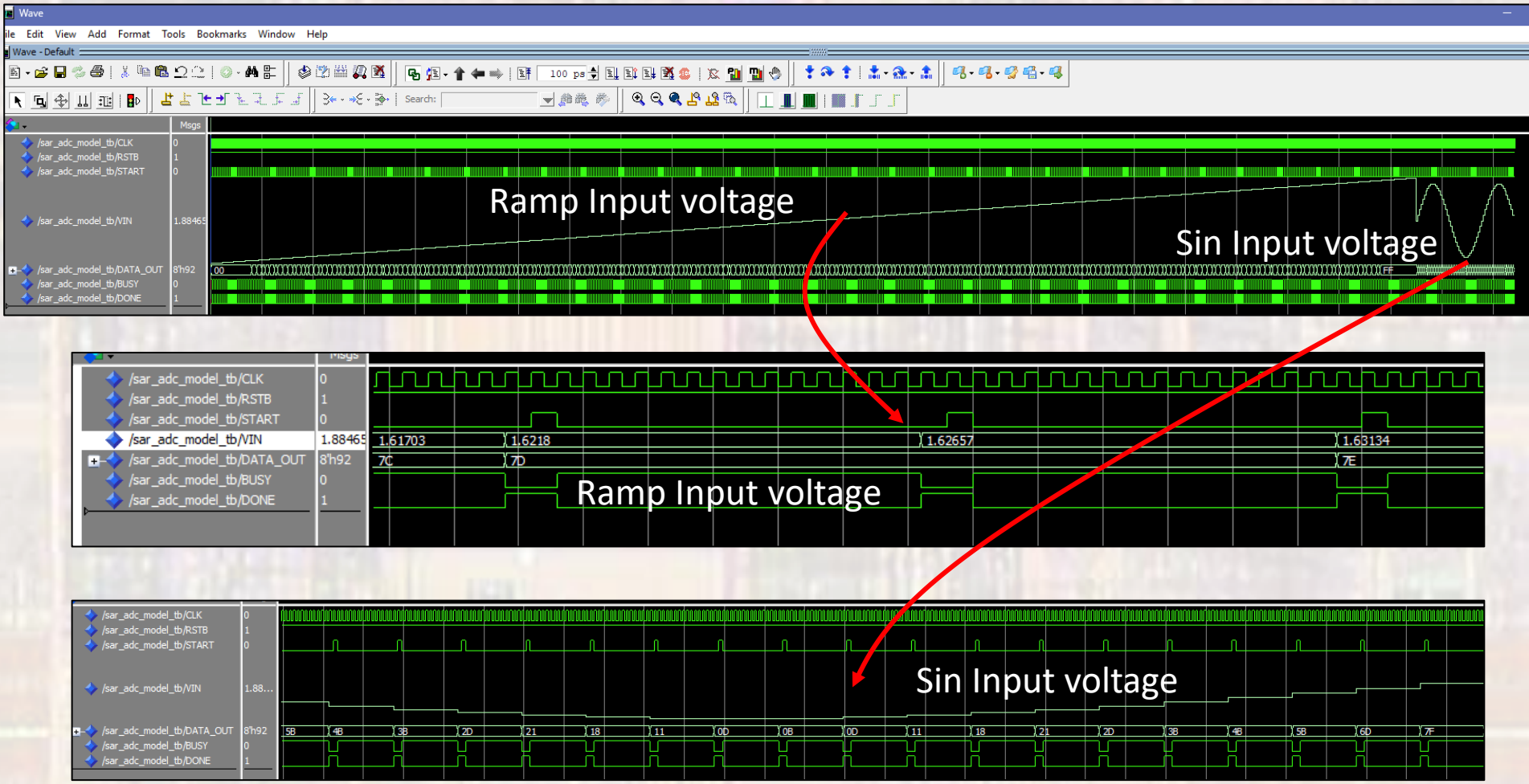
- SAR Type ADC Model – RTL
  - No RTL possible since many blocks are not synthesizable
  - Questa can draw a diagram



 © tj

ADC – SAR Model

- SAR Type ADC Model – Simulation



 © tj