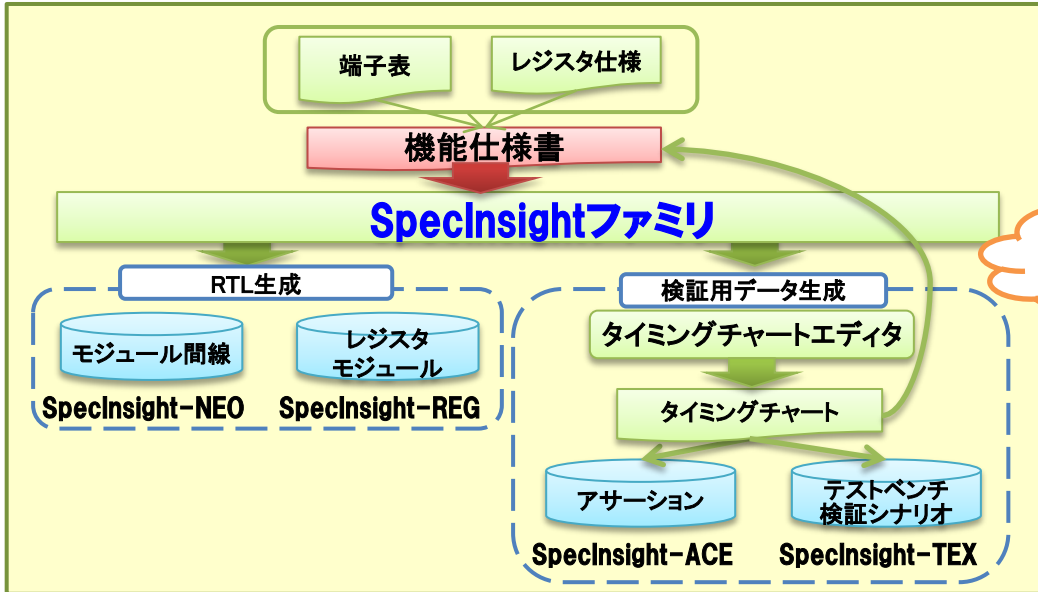




SpecInsight

仕様書情報から設計検証データを自動生成！



SpecInsight-NEO (モジュール間結線生成ツール)

- 【機能】**
- 入出力端子表(図1)と接続情報からモジュール間接続RTL(図2)を自動生成します
 - RTL記述言語は、Verilog または VHDL から選択可能です
- 【特徴】**
- 入出力端子表、接続情報はExcelファイルで作成可能です
 - お客様のフォーマットをそのまま使用することもできます
 - ビット幅の不一致や入出力定義の不整合をチェックできます
- 【効果】**
- 面倒で単純ミスが発生しやすいモジュール間接続作成から解放されます

| AXIバス | 信号名 | 方向 | タイプ | ビット幅 | 初期値 |
|-------|-----------|----|-----|------|-----|
| | A0_WID | I | clk | - | 8 |
| | A0_MDATA | I | clk | - | 64 |
| | A0_MLAST | I | clk | P | 1 |
| | A0_MSTRB | I | clk | - | 8 |
| | A0_MREADY | O | clk | P | 1 |
| | A0_BID | I | clk | - | 8 |
| | A0_BRESP | I | clk | - | 2 |
| | A0_BVALID | I | clk | - | 1 |
| | A0_BREADY | O | clk | P | 1 |
| | A0_ARID | I | clk | - | 8 |
| | A0_ARADDR | I | clk | - | 32 |
| | A0_ARLEN | I | clk | - | 5 |
| | A0_ARSIZE | I | clk | - | 3 |

図1 入出力端子表の例

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;

entity ai is
port (
  clk      : in  std_logic; -- クロック
  rst_x    : in  std_logic; -- ハードウェアリセット
  A1_AWID  : in  std_logic_vector ( 7 downto 0 );
  A1_ADDR  : in  std_logic_vector ( 31 downto 0 );
  A1_MLEN  : in  std_logic_vector ( 4 downto 0 );
  A1_AWSIZE : in  std_logic_vector ( 2 downto 0 );
  A1_AWBURST : in  std_logic_vector ( 2 downto 0 );
  A1_MLOCK : in  std_logic;
  ..
);
```

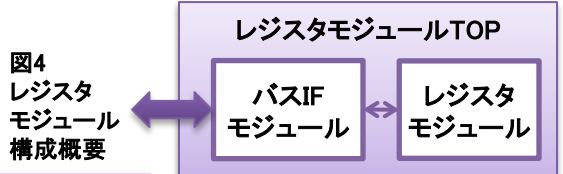
図2 RTL出力例

SpecInsight-REG (レジスタモジュール生成ツール)

- 【機能】**
- レジスタ仕様(図3)からレジスタモジュールRTL(図4)を自動生成します
 - RTL記述言語は、Verilog または VHDL から選択可能です
 - バスインタフェースは標準バスをサポートします
- 【特徴】**
- レジスタ名の重複やアドレスマップのミスをチェックできます
 - AXI, AHB, APBバスに対応します
- 【効果】**
- レジスタ数が多くても、レジスタ仕様を定義すれば簡単にRTLを生成できます
 - 仕様書やRTLの形式を統一することができます

| ヘッアドレスグループ | グループサイズ (byte) | グループ名 (ファイル名) | ビットアサイン | バスアクセス属性 | Reset初期値 | 接続内部IF |
|------------|----------------|---------------|---------|----------|----------|--------|
| 0x000 | 0x100 | UART | | | | |
| 0x00 | | Data | [7:0] | RW | - | RWfifo |
| 0x04 | | Status | | | | |
| | | TxFULL | [0] | RO | | I |
| | | RxVALID | [1] | RO | | I |

図3 レジスタ仕様の例



【ムービーで見る】 https://www.youtube.com/watch?v=QF1rj7Z9Bus&feature=player_embedded

SpecInsight-ACE (アサーション自動生成ツール)

【機能】

- ・専用エディタで仕様書(タイミングチャート)を作成できます(図5)
- ・タイミングチャート上にアサーション仕様を図で定義できます(図6)
- ・アサーション仕様からSystemVerilog Assertion コード(図7)を生成します

【特徴】

- ・タイミングチャートを簡単に作成することができます
- ・アサーション言語を知らなくてもアサーションを定義できます
- ・図で定義したアサーションを日本語の説明文として出力します

【効果】

- ・アサーション・ベース検証を簡単に導入することができます
- ※アサーションに関する詳しい知識は必要ありません
- ・検証内容を図で表示できるので、レビューがしやすくなります
- ・タイミングチャートもアサーションも簡単に再利用できます

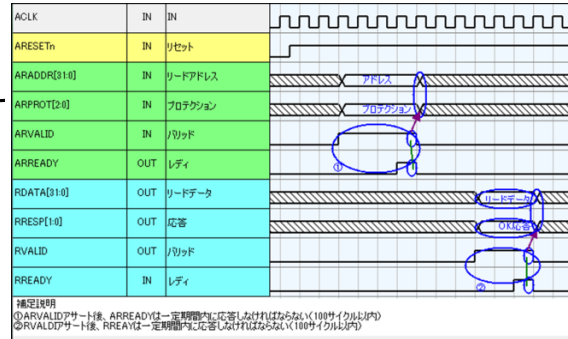


図5 タイミングチャート例(仕様書用)



図6 アサーション定義例

```

アサーショングループ名 READ_ADR
アサーション名 AXI4_LiteS_ARREADY_MAX_WAIT
アサーション説明 ARVALIDとARREADYの応答時間
アサーション内容 ARVALIDが0から1に変化した時、0から100サイクル後、(ARREADY=1)とならなければならない

property p_axi4_lite_s_arready_max_wait;
  @(posedge ACLK)
  ((spast(ARVALID) == 0) && (!stable(ARVALID))) && (ARVALID == 1) |>
  ##[0:100] (ARREADY == 1);
endproperty

AXI4_LiteS_ARREADY_MAX_WAIT: assert property(p_axi4_lite_s_arready_max_wait)
else begin
  $display("##### [ ERROR ] AXI4_LiteS_ARREADY_MAX_WAIT");
end
    
```

図7 アサーション・コードの出力例

【ムービーで見る】タイミングチャート作成 https://www.youtube.com/watch?feature=player_embedded&v=GnHS8gSLe6k
 アサーション作成 https://www.youtube.com/watch?feature=player_embedded&v=Coe5I-84u_s

SpecInsight-TEX (テストベンチ生成ツール)

【機能】

- ・仕様書用に作成したタイミングチャートから、テストベンチの入力情報を生成します
- ・複数のタイミングチャートを組み合わせて検証シナリオを作成でき、テストベンチと一緒に出力できます(図8)
- ・テストベンチ記述言語は、Verilog または VHDL から選択可能です

【特徴】

- ・タイミングチャートがあれば、簡単な操作でテストベンチを作成できます
- ・RTLとテストベンチは入出力端子表から自動で接続します
- ・SpecInsight-ACEで生成したアサーションも簡単に取り込めます

【効果】

- ・テストベンチに不慣れな人でも簡単にシミュレーションを実行できます
- ・設計の初期段階で簡単にシミュレーションによる動作確認ができます

```

entity TB_TOP is
  and entity;
architecture RTL of TB_TOP is
  component apb
  port(
    PCLK : in std_logic;
    PRESET : in std_logic;
    PADDR : in std_logic_vector(15 downto 0);
    PWRITE : in std_logic;
    PENABLE : in std_logic;
    PSEL : in std_logic;
    PRDATA : in std_logic_vector(31 downto 0);
    PRDATA : out std_logic_vector(31 downto 0);
    PREADY : out std_logic;
    GPIN0 : in std_logic_vector(15 downto 0);
    GPIN1 : in std_logic_vector(15 downto 0);
    GPOUT0 : out std_logic_vector(15 downto 0);
    GPOUT1 : out std_logic_vector(15 downto 0);
    SCLK : in std_logic;
    SIN_EN : in std_logic;
    SIN : in std_logic;
    SOUT_EN : out std_logic;
    SOUT : out std_logic;
  );
end component;
    
```

```

procedure APB_Slave_Access_proc ( signal tb_sig : inout tb_signal) is
  variable APB_Write_PADDR : std_logic_vector(15 downto 0);
  variable APB_Write_PRDATA : std_logic_vector(31 downto 0);
  variable APB_Read_PADDR : std_logic_vector(15 downto 0);
  variable APB_Read_PRDATA : std_logic_vector(31 downto 0);
begin
  tb_sig.PRESET <= '0';
  tb_sig.PADDR <= (others >= '0');
  tb_sig.PWRITE <= '0';
  tb_sig.PSEL <= '0';
  tb_sig.PENABLE <= '0';
  tb_sig.PRDATA <= (others >= '0');
  wait until tb_sig.PCLK'event and tb_sig.PCLK = '1';
  wait until tb_sig.PCLK'event and tb_sig.PCLK = '1';
  wait until tb_sig.PCLK'event and tb_sig.PCLK = '1';
  wait until tb_sig.PCLK'event and tb_sig.PCLK = '1';
  wait until tb_sig.PCLK'event and tb_sig.PCLK = '1';
  wait until tb_sig.PCLK'event and tb_sig.PCLK = '1';
  wait until tb_sig.PRESET <= '1';
  wait until tb_sig.PCLK'event and tb_sig.PCLK = '1';
  APB_Write_PADDR := std_logic_vector(conv_unsigned(1688010H, 16));
  APB_Write_PRDATA(30 downto 0) := std_logic_vector(conv_unsigned(16H12345678H, 31));
  APB_Write_PRDATA(31 downto 31) := std_logic_vector(conv_unsigned(1680H, 1));
  APB_Write(tb_sig => tb_sig, O_PADDR => APB_Write_PADDR, O_PRDATA => APB_Write_PRDATA);
  wait until tb_sig.PCLK'event and tb_sig.PCLK = '1';
    
```

図8 タイミングチャートとテストベンチ出力例

【ムービーで見る】 https://www.youtube.com/watch?v=5GlsIE56SF0&feature=player_embedded

| 機能 | 製品名 | 備考 |
|-------------|-----------------------|-------------------------------|
| RTL生成 | SpecInsight-NEO | |
| | SpecInsight-REG | |
| アサーション生成 | SpecInsight-ACE | |
| テストベンチ生成 | SpecInsight-TEX | (*) タイミングチャートエディタ単体もご購入いただけます |
| タイミングチャート作成 | SpecInsight-タイミングチャート | |



各製品は、予告なく変更する場合があります
2015年5月 Rev1.4