

HDL テキストエディタ

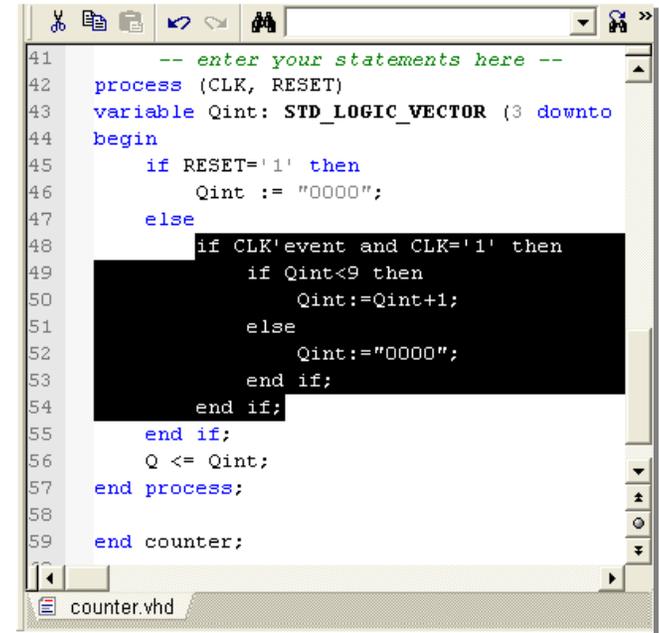
Active-HDL アプリケーションノート

HDL エディタ概要

- Active-HDL の HDL エディタには下記機能があります。
 - ブロック/矩形の選択、コメント化、コメント解除
 - TABインデント、逆TABインデント
 - オートコンプリート機能
 - 文字列検索、置換
 - シンタックスハイライト
 - ブックマーク
 - オートインデント/スマートインデント
 - オートフォーマット
 - 構造解析モード
 - 言語アシスタント
 - 外部テキストエディタサポート
 - アクションレコーダー/プレーヤー

ブロック選択

- マウスによる選択方法: 選択するブロックの先頭にマウスでカーソルを配置。左ボタンを押しながら矢印キーでカーソルを選択するブロックの終点に移動。
- キーボードでの選択方法: 選択するブロックの先頭に矢印キーでカーソルを配置。**Shift** キーを押しながら矢印キーでカーソルを選択するブロックの終点に移動。



```
41 -- enter your statements here --
42 process (CLK, RESET)
43 variable Qint: STD_LOGIC_VECTOR (3 downto
44 begin
45     if RESET='1' then
46         Qint := "0000";
47     else
48         if CLK'event and CLK='1' then
49             if Qint<9 then
50                 Qint:=Qint+1;
51             else
52                 Qint:="0000";
53             end if;
54         end if;
55     end if;
56     Q <= Qint;
57 end process;
58
59 end counter;
```

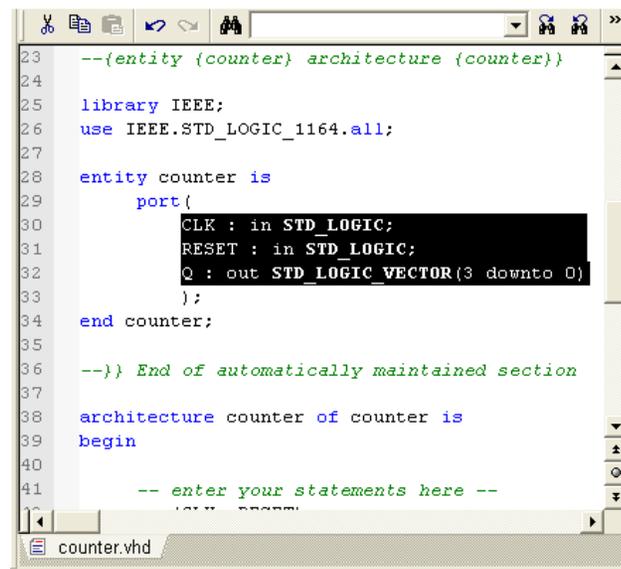
Note:

単語選択: **Ctrl + Shift** キーを押しながら特定の単語をマウスでクリックすると単語全体を選択することができます。

選択色の変更: Preferences画面 | Fonts and Colorsタブ | Selection で選択した背景色を変更可能です。Preferences画面表示方法: メニュー | View>HDE Optionを選択、または、HDLエディタを右クリックし、ポップアップメニューからPreferences を選択。

矩形選択

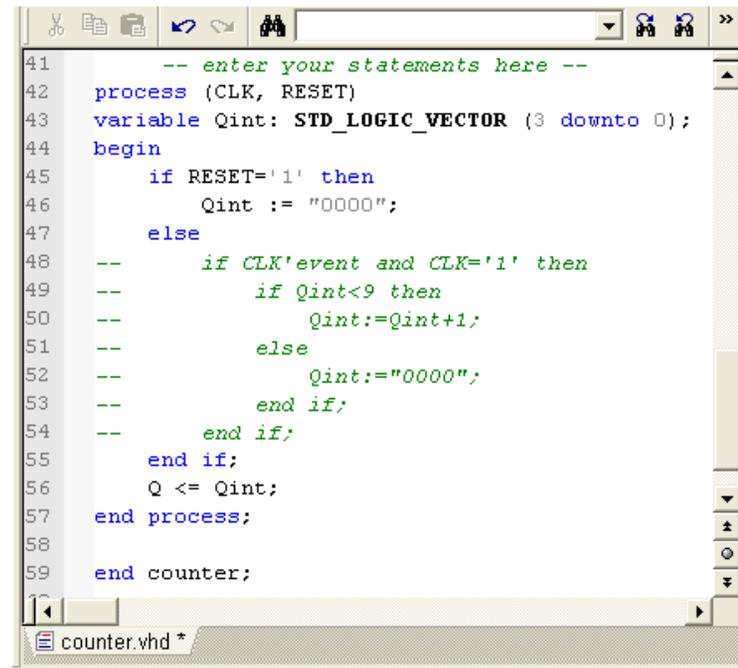
- マウスによる選択方法: 選択するブロックの先頭にマウスでカーソルを配置。Alt キー + 左ボタンを押しながら矢印キーでカーソルを選択するブロックの終点に移動。
- キーボードでの選択方法: ツールボタン  を選択、または Alt + C を押す。そして選択するブロックの先頭に矢印キーでカーソルを配置。Shift キーを押しながら矢印キーでカーソルを選択するブロックの終点に移動。



```
23  --{entity {counter} architecture {counter}}
24
25  library IEEE;
26  use IEEE.STD_LOGIC_1164.all;
27
28  entity counter is
29      port(
30          CLK : in STD_LOGIC;
31          RESET : in STD_LOGIC;
32          Q : out STD_LOGIC_VECTOR(3 downto 0)
33      );
34  end counter;
35
36  --}) End of automatically maintained section
37
38  architecture counter of counter is
39  begin
40
41      -- enter your statements here --
42  end;
```

ブロックのコメント化

- コメントにしたいブロックを選択し、下記のいずれかの方法でコメント化することができます。
 - ツールボタン  をクリック
 - エディタ上を右クリック | ポップアップメニュー | Block | Comment を選択
 - Ctrl+K

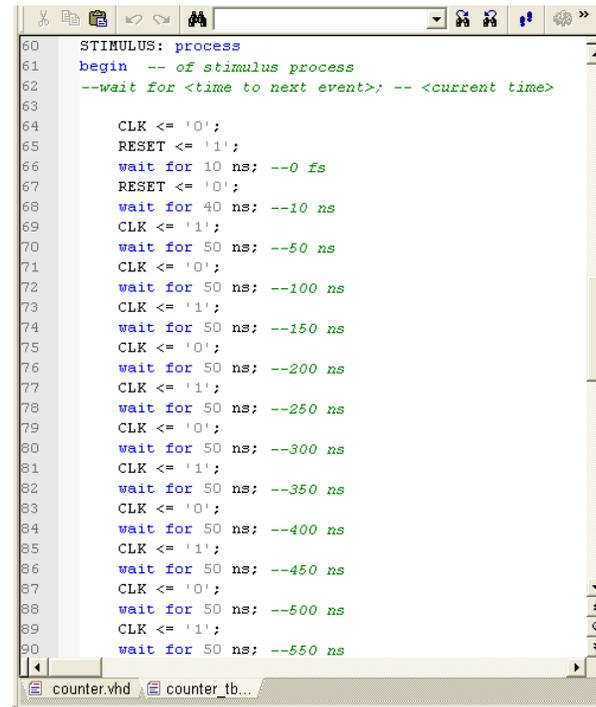


```
41      -- enter your statements here --
42  process (CLK, RESET)
43  variable Qint: STD_LOGIC_VECTOR (3 downto 0);
44  begin
45      if RESET='1' then
46          Qint := "0000";
47      else
48          --      if CLK'event and CLK='1' then
49          --          if Qint<9 then
50          --              Qint:=Qint+1;
51          --          else
52          --              Qint:="0000";
53          --          end if;
54          --      end if;
55      end if;
56      Q <= Qint;
57  end process;
58
59  end counter;
```

counter.vhd *

矩形のコメント化

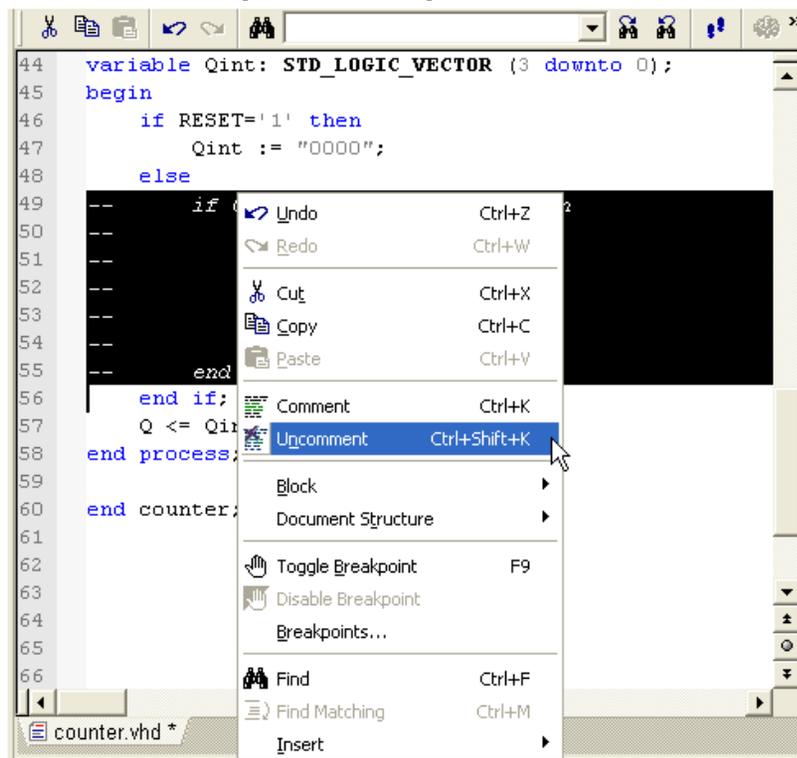
- 下図のような場合には矩形を選択して、コメント化すると便利です。
 - ツールボタン  をクリック
 - エディタ上を右クリック | ポップアップメニュー | Block | Comment を選択
 - Ctrl+K



```
60 STIMULUS: process
61 begin -- of stimulus process
62 --wait for <time to next event>; -- <current time>
63
64     CLK <= '0';
65     RESET <= '1';
66     wait for 10 ns; --0 fs
67     RESET <= '0';
68     wait for 40 ns; --10 ns
69     CLK <= '1';
70     wait for 50 ns; --50 ns
71     CLK <= '0';
72     wait for 50 ns; --100 ns
73     CLK <= '1';
74     wait for 50 ns; --150 ns
75     CLK <= '0';
76     wait for 50 ns; --200 ns
77     CLK <= '1';
78     wait for 50 ns; --250 ns
79     CLK <= '0';
80     wait for 50 ns; --300 ns
81     CLK <= '1';
82     wait for 50 ns; --350 ns
83     CLK <= '0';
84     wait for 50 ns; --400 ns
85     CLK <= '1';
86     wait for 50 ns; --450 ns
87     CLK <= '0';
88     wait for 50 ns; --500 ns
89     CLK <= '1';
90     wait for 50 ns; --550 ns
```

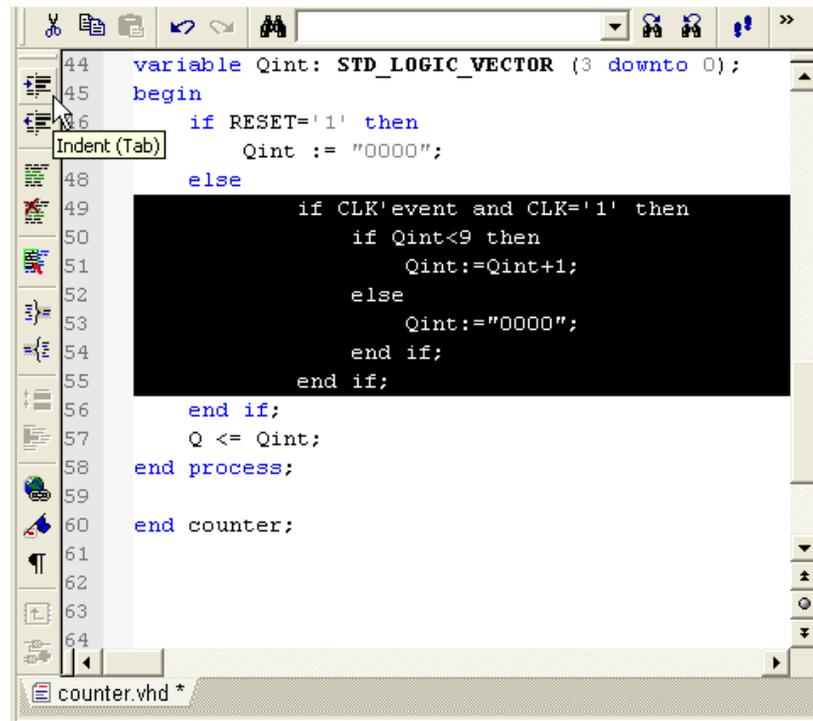
ブロック/矩形のコメント解除

- ブロックや矩形を選択し、下記のいずれかの方法でコメントを解除することができます。
 - ツールボタン  をクリック
 - エディタ上を右クリック | ポップアップメニュー | Block | Uncomment を選択
 - Ctrl+Shift+K



ブロックのTABインデント

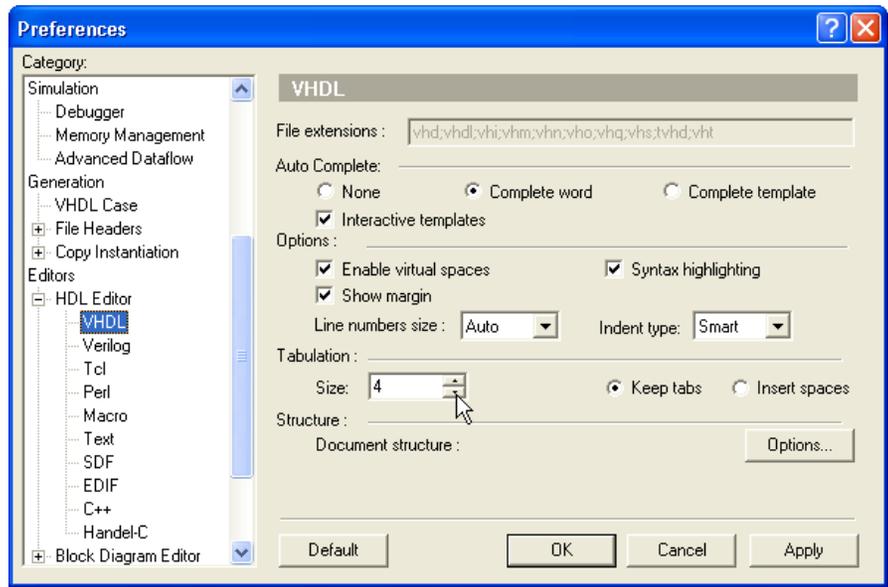
- ブロックを選択し、下記のいずれかの方法でTABでインデントすることができます。
 - ツールボタン  をクリック
 - エディタ上を右クリック | ポップアップメニュー | Block | Indent を選択
 - TAB



```
44 variable Qint: STD_LOGIC_VECTOR (3 downto 0);
45 begin
46     if RESET='1' then
47         Qint := "0000";
48     else
49         if CLK'event and CLK='1' then
50             if Qint<9 then
51                 Qint:=Qint+1;
52             else
53                 Qint:="0000";
54             end if;
55         end if;
56     end if;
57     Q <= Qint;
58 end process;
59
60 end counter;
```

ブロックの逆TABインデント

- ブロックを選択し、下記のいずれかの方法で逆方向にTABでインデントすることができます。
 - ツールボタン  をクリック
 - エディタ上を右クリック | ポップアップメニュー | Block | Outdent を選択
 - Shift+TAB

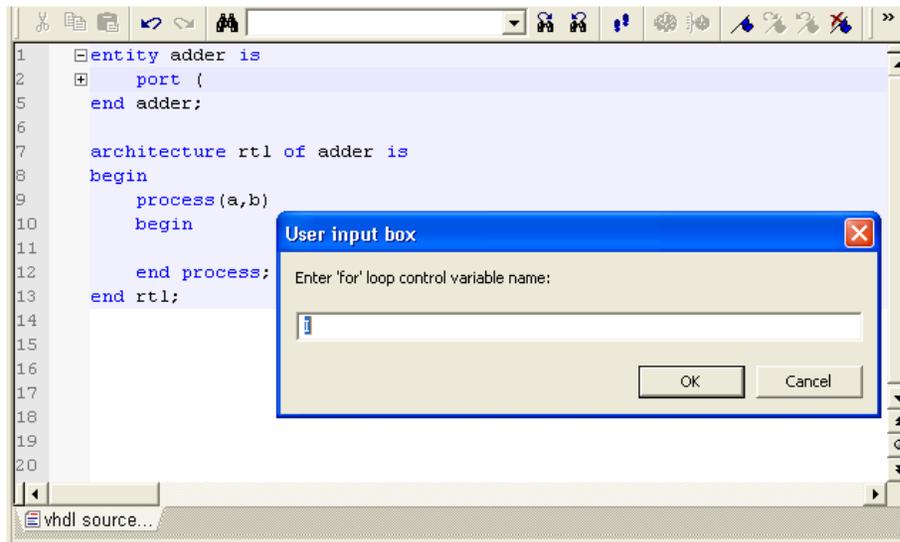


Note: TAB はデフォルトで 4 に設定されています。Preferences 画面 | HDL Editor タブ(上図)の Size で任意の値に変更可能です。

オートコンプリート機能

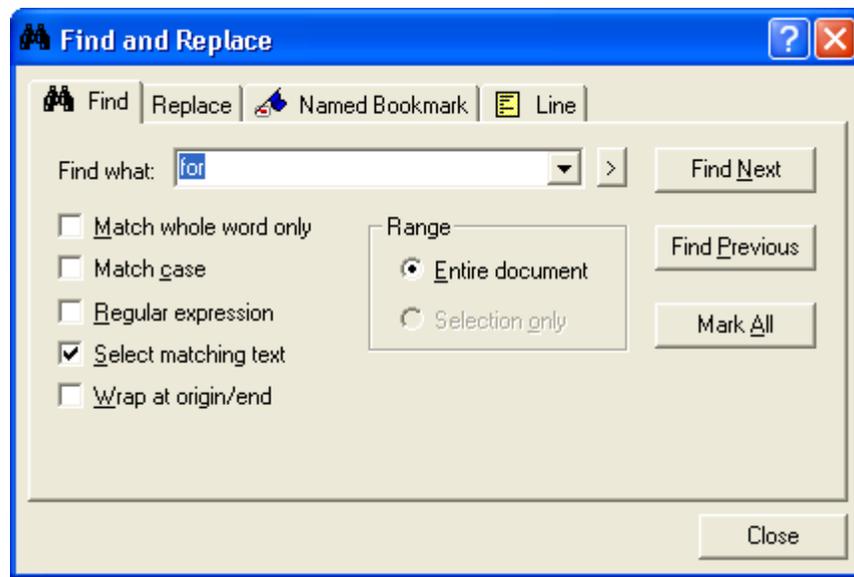
- VHDL, Verilog キーワードを初めの数文字打ち込むことにより補完
 - TAB キーによってキーワードの切り替え
- インタラクティブテンプレート
 - module, entity and architecture 等の補完を表示している際に Ctrl+Enter を押すとインタラクティブテンプレートを挿入

```
1  entity adder is
2  port (
5  end adder;
6
7  architecture
```



文字列検索

- 検索ウィンドウ表示
 - メニュー | Search | Find
 - Ctrl+F

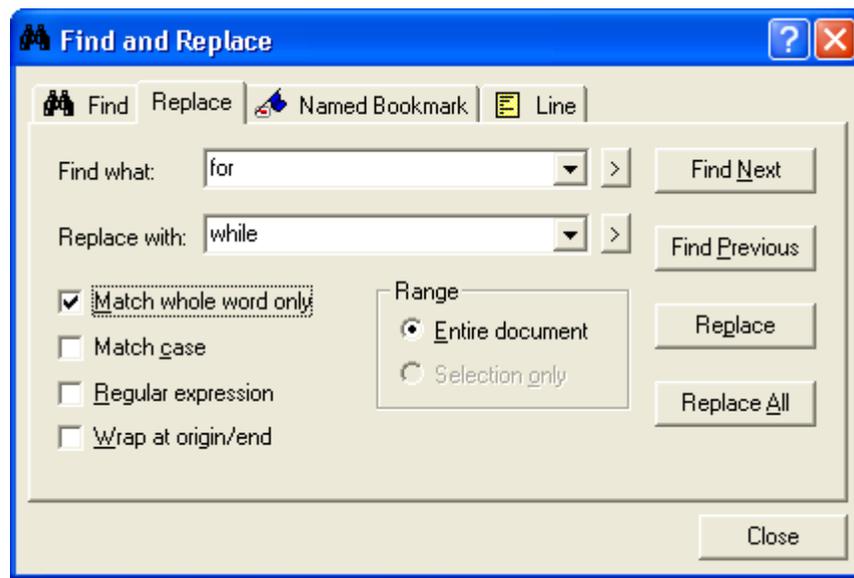


Note: 複数ファイルからの文字列検索

- メニュー | Search | Find in files

文字列置換

- 置換ウィンドウ表示
 - メニュー | Search | Replace
 - Ctrl+H



シンタックスハイライト

- 下記ファイルのシンタックスハイライトが可能です

- VHDL

```
entity adder is
    generic ( tpd : time )
    port ( inp : in bit; outp : out bit );
end entity adder;
```

- Verilog

- EDIF

- SDF

- C/C++

- Tcl/Tk

- Perl

- .DO macro

```
always @(posedge clk) begin
    if (!en)
        out1 <= 1'bz;
end
```

```
(interface
    (port IO
        (direction INPUT)
        (property PINTYPE ( string "IN" ) )
        (property port_id ( string "3" ) )
    )
)
```

```
(SETUP (posedge I) (posedge CLK) (10:20:30))
(HOLD (posedge I) (posedge CLK) (0:0:0))
(SETUP (negedge I) (posedge CLK) (10:20:30))
(HOLD (negedge I) (posedge CLK) (0:0:0))
```

```
// check 1
if (vhpi_check_error(&vhpiErrorInfo)){
    switch ( vhpiErrorInfo.severity ){
        case vhpiNote : break;
        case vhpiWarning : break;
    }
```

```
proc ExamineAll () {
    global hold bust d_l d_h DisplayTopic;
    set hold [examine HOLD];
    set bust [examine BUST];
}
```

```
if (++$numopen > $maxopen) {
    splice(@lru, $maxopen / 3);
    $numopen -= @lru;
    for (@lru) { &close($_); delete $isopen($_); }
```

```
comp -include "$DSN\src\TestBench\counter_TB.vhd"
asim TESTBENCH_FOR_counter
wave
wave -noreg CLK
```

ナビゲーション

- HDL エディタの各種機能を使用してソースコードのナビゲーションを容易に実行することができます。

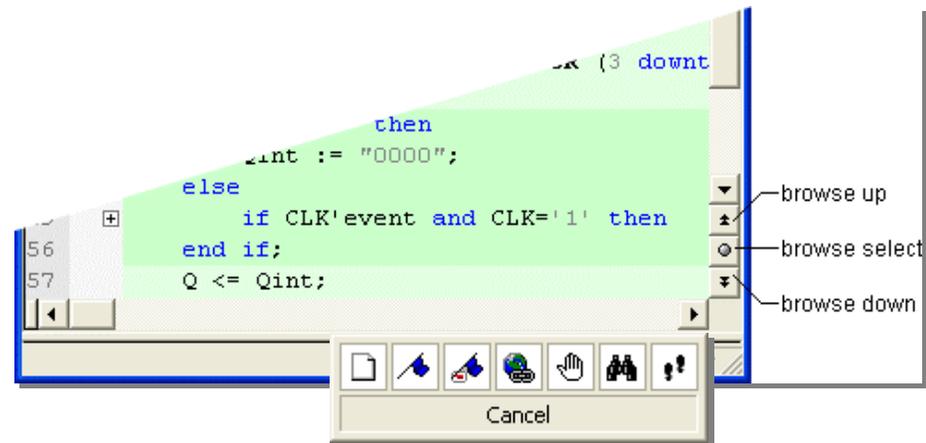
- ブックマーク
- 名称ブックマーク
- リンク
- ブラウズボタン
- 画面分割

- ブラウズボタンによって下記の方法でスクロールできます。

- ページ
- リンク
- ブックマーク
- ブレークポイント
- 名称ブックマーク

```
29 entity counter is
30   port(
31     CLK : in STD_LOGIC;
32     RESET : in STD_LOGIC;
33     Q : out STD_LOGIC_VECTOR
34   );
35 end counter;

43 process (CLK, RESET)
44   variable Qint: STD_LOGIC_VECTOR (
45   begin
46     if RESET='1' then
47       Qint := "0000";
48     else
49       if CLK'event and CLK='1' then
50         Qint := Q;
51       end if;
52     end if;
53     Q <= Qint;
54   end process;
55 end counter;
```



ブックマーク

- ブックマークの設定方法
 - ツールボタン  をクリック
 - Ctrl+F2 キー
- ブックマークのジャンプ方法
 -  ボタン: 次のブックマークへ
 -  ボタン: 前のブックマークへ
- 全てのブックマークの解除
 - ツールボタン  をクリック

```
45 begin
46  if RESET='1' then
47     Qint := "0000";
48 else
49     if CLK'event and CLK='1' then
50  if Qint<9 then
51         Qint:=Qint+1;
52     else
53         Qint:="0000";
```

Search	View	Workspace	Design	Simu
	Find...			Ctrl+F
	Replace...			Ctrl+H
	Find in Files...			Ctrl+Shift+F
	Replace in Files...			
<hr/>				
	Goto...			Ctrl+G
<hr/>				
	Next Message			F4
	Previous Message			Shift+F4
<hr/>				
	Toggle Bookmark			Ctrl+F2
	Next Bookmark			F2
	Previous Bookmark			Shift+F2
	Clear All Bookmarks			

Note: メニュー | Search から同様の操作が行えます。

名称ブックマーク

- 名称ブックマークでは通常のブックマークの変わりにソースコード中に自分で決めた名前のブックマークを挿入することができます。ブックマークに名前を付けておきたい場合などに便利です。
- 名称ブックマークの設定方法
 - ツールボタン  をクリック
 - エディタ上を右クリック | Insert | Named Bookmark
- 名称ブックマークのジャンプ方法
 - ブラウズボタンで名称ブックマークを選択
 - ブラウズボタン上下にボタンで移動

オートインデント/スマートインデント

- Preferences 画面でオートインデントかスマートインデントの設定が可能です。

- オートインデント

- Enter キーで改行の際に前の行の先頭と同じ位置にタブ/スペースを設定

```
72     CLK <= '0';
73     ADDR <= "00000000";
74     IDATA <= "00000000";
75     CS <= '1';
76     WE <= '0';
77     RST <= '1';
78     wait for 5 ns; --0 fs
79     CLK <= '1';
80     wait for 5 ns; --5 ns
81     CLK <= '0';
82     wait for 5 ns; --10 ns
83     CLK <= '1';
84     wait for 5 ns; --15 ns
85     CLK <= '0';
86     wait for 5 ns; --20 ns
87     CLK <= '1';
88     wait for 5 ns; --25 ns
89     CLK <= '0';
90     wait for 5 ns; --30 ns
91     CLK <= '1';
92     wait for 5 ns; --35 ns
93     CLK <= '0';
94     wait for 5 ns; --40 ns
95     CLK <= '1';
96     wait for 5 ns; --45 ns
97     CLK <= '0';
```

- スマートインデント

- Enter キーで改行すると HDL 構文にあわせてタブ/スペースを設定

```
architecture fifo_reader_arch of fifo_reader is
begin
    process(CLK_IN, RESET)
    begin
        if RESET='1' then
            DATA_OUT<="ZZZZZZZZ";
            READ_FIFO<='0';
        elsif CLK_IN='1' and CLK_IN'event then
            if PAUSE='0' and READ_DATA='1' then
                READ_FIFO<='1';
                DATA_OUT<=DATA_IN;
            else
                READ_FIFO<='0';
            end if;
        end if;
    end process;
end architecture;
```

オートフォーマット

- ツールボタン  によってソースコードの自動整形ができます。ソースコードを解析し、スマートインデントと同じ形に自動整形。

```
entity fifo_reader is
port (
  RESET : in STD_LOGIC;
  CLK_IN : in STD_LOGIC;
  DATA_IN : in STD_LOGIC_VECTOR (7 downto 0);
  PAUSE : in STD_LOGIC;
  READ_DATA : in STD_LOGIC;
  READ_FIFO : out STD_LOGIC;
  DATA_OUT : out STD_LOGIC_VECTOR (7 downto 0)
);
end entity;

architecture fifo_reader_arch of fifo_reader is
begin

process(CLK_IN,RESET)
begin
  if RESET='1' then
    DATA_OUT<="ZZZZZZZZ";
    READ_FIFO<='0';
  elsif CLK_IN='1' and CLK_IN'event then
    if PAUSE='0' and READ_DATA='1' then
      READ_FIFO<='1';
      DATA_OUT<=DATA_IN;
    else
      READ_FIFO<='0';
    end if;
  end if;
end process;

end architecture;
```

ビフォー



アフター

```
entity fifo_reader is
  port (
    RESET : in STD_LOGIC;
    CLK_IN : in STD_LOGIC;
    DATA_IN : in STD_LOGIC_VECTOR (7 downto 0);
    PAUSE : in STD_LOGIC;
    READ_DATA : in STD_LOGIC;
    READ_FIFO : out STD_LOGIC;
    DATA_OUT : out STD_LOGIC_VECTOR (7 downto 0)
  );
end entity;

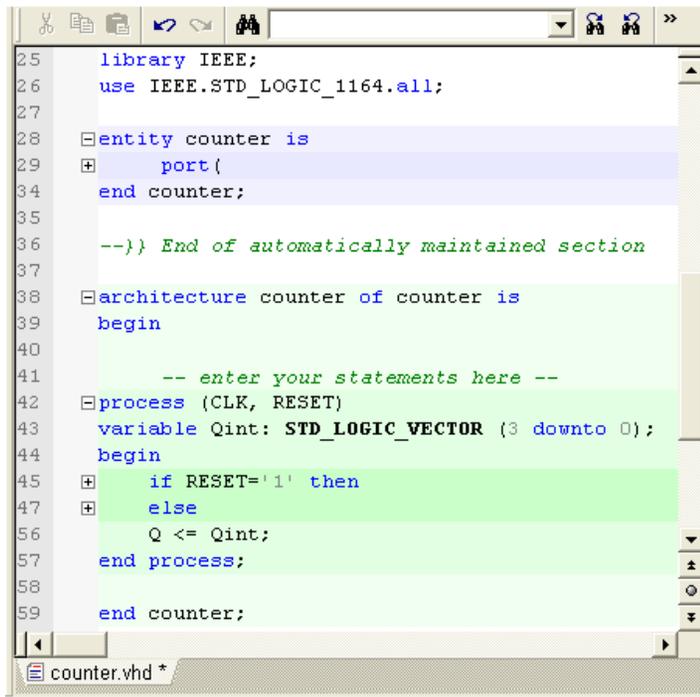
architecture fifo_reader_arch of fifo_reader is
begin

  process(CLK_IN,RESET)
  begin
    if RESET='1' then
      DATA_OUT<="ZZZZZZZZ";
      READ_FIFO<='0';
    elsif CLK_IN='1' and CLK_IN'event then
      if PAUSE='0' and READ_DATA='1' then
        READ_FIFO<='1';
        DATA_OUT<=DATA_IN;
      else
        READ_FIFO<='0';
      end if;
    end if;
  end process;

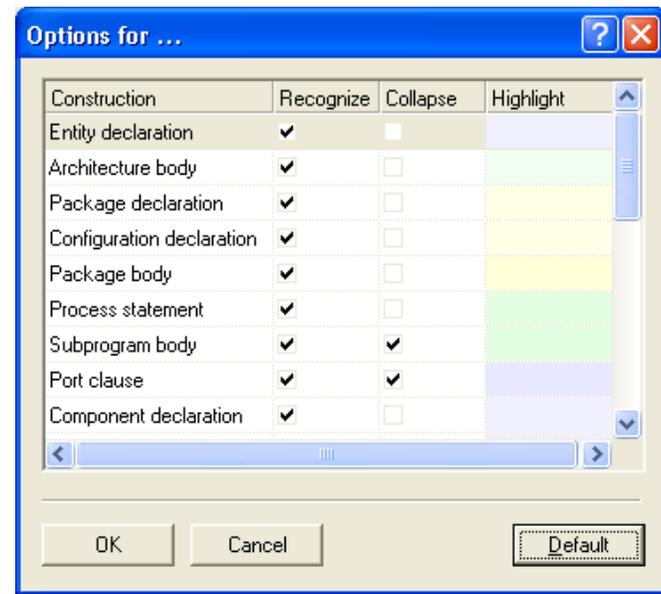
end architecture;
```

構造解析モード

- ツールボタン  によって HDL シンタックスルールに従い、開閉可能なグループに分割



```
25 library IEEE;
26 use IEEE.STD_LOGIC_1164.all;
27
28 entity counter is
29     port(
30         end counter;
31
32     --}) End of automatically maintained section
33
34 architecture counter of counter is
35     begin
36         -- enter your statements here --
37
38     process (CLK, RESET)
39         variable Qint: STD_LOGIC_VECTOR (3 downto 0);
40         begin
41             if RESET='1' then
42             else
43                 Q <= Qint;
44             end process;
45
46     end counter;
```



Note: Preferences ウィンドウでグループ化するHDL構文の選択やグループ化の色の設定が可能

構造解析モード

- 構造解析モードで   ボタンをクリックするとグループの展開、格納が可能
- 独自のグループ化が可能。ブロックを選択し、 ボタンをクリック
- ツールボタン  をクリックすると構造解析モードが解除され、通常レイアウトに戻る。

```
1      library IEEE;
2      use IEEE.std_logic_1164.all;
3
4      entity fifo_reader is
5      +   port (
14     end entity;
15
16     architecture fifo_reader_arch of fifo_reader is
17     begin
18
19     process(CLK_IN,RESET)
20     begin
21     +   if RESET='1' then
24     +   elsif CLK_IN='1' and CLK_IN'event then
25     +   if PAUSE='0' and READ_DATA='1' then
28     -   else
29         READ_FIFO<='0';
30     end if;
31     end if;
32     end process;
33
34     end architecture;
35
36     begin
37
38     ----- Component instantiations -----
39
40     U1 : fifo
41     port map(
42         RE => RE,
43         RF => RF,
44         CE => CE,
45         CLK => CLK_FIFO,
46         CLR => CLR,
47         CLRptr => CLRptr,
48         DATA => DATA_IN,
49         Q => DATA,
50         RD => RD,
51         WR => WR
52     );
53
54     U2 : fifo_reader
55     port map(
```

言語アシスタント

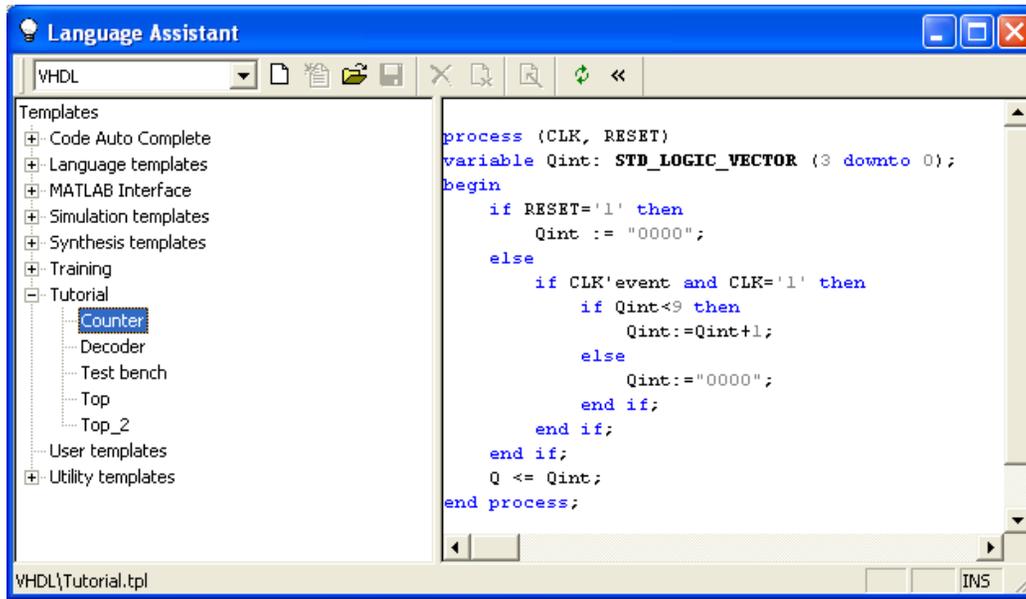
- 言語アシスタント機能は、VHDL、Verilog、アサーション言語などの作成をアシストします。多くの記述テンプレートが用意されています。下記のテンプレートから構成されています。
 - 言語テンプレート: ベーシックな言語構文
 - シンセシス・テンプレート: マルチプライヤー、フリップフロップ、カウンタなどのファンクションブロック
 - シミュレーション・テンプレート: テストベンチ作成に役立つ構文
 - オートコンプリート: オートコンプリート機能の為のテンプレート。テンプレートに単語を登録可能。
 - マクロコマンドテンプレート: ベーシックなマクロコマンドテンプレート

Note: 言語アシスタント画面表示方法

- ツールボタン  をクリック
- メニュー | Tools | Language Assistant

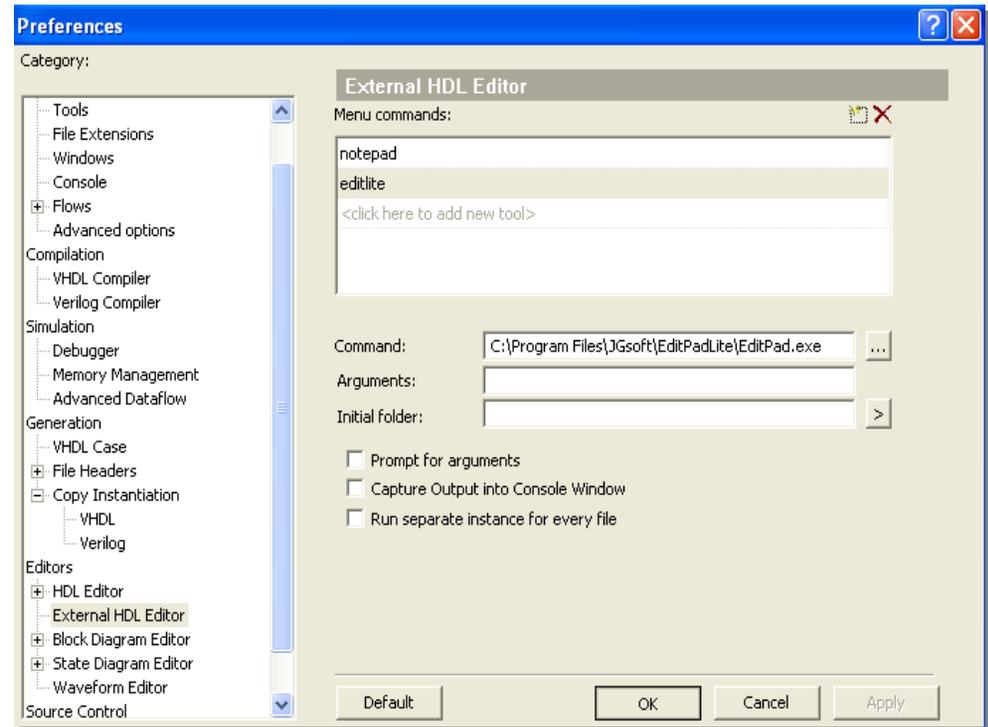
言語アシスタント

- テンプレートの HDL エディタへの挿入
 - テンプレート名を HDL エディタヘドラック&ドロップ
 - 挿入したい場所にカーソルを合わせテンプレート名を右クリック | ポップアップメニュー | Use



外部テキストエディタサポート

- Active-HDLでは外部のエディタを登録し、使用することができます。
 - メニュー | Tools | 」Preferences | External HDL Editor ウィンドウを表示
 - エディタ名を入力
 - 実行ファイルのディレクトリを指定

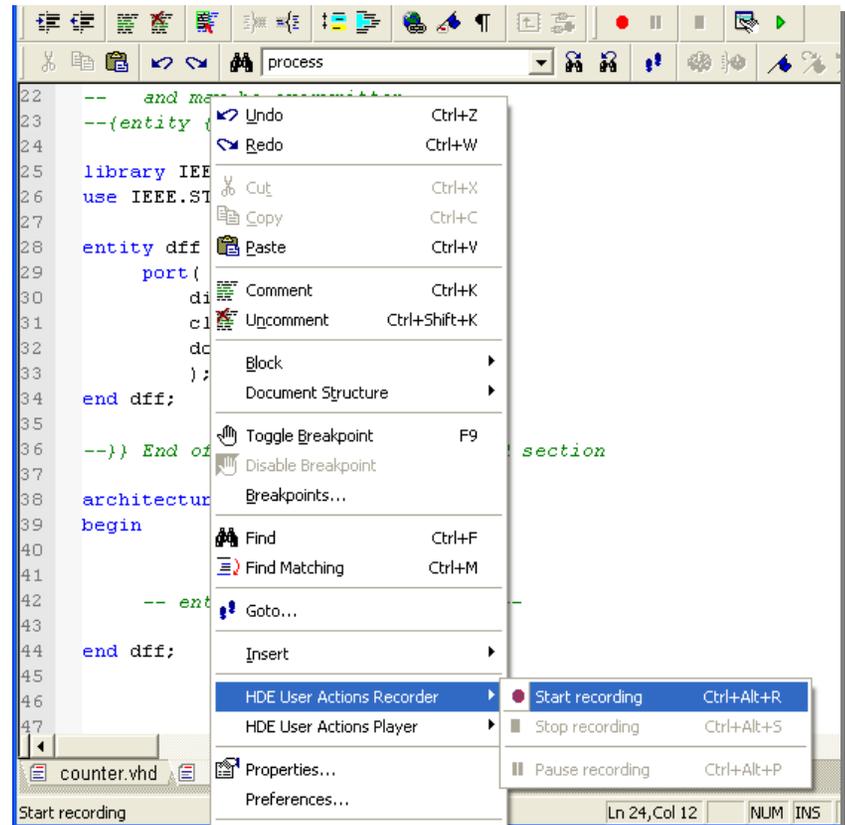


外部テキストエディタサポート

- 登録した外部エディタの起動方法
 - デザインブラウザ | Files タブ | ソースコードファイルを右クリック | Open in External Editor から登録した外部エディタを選択

アクションレコーダー/プレーヤー

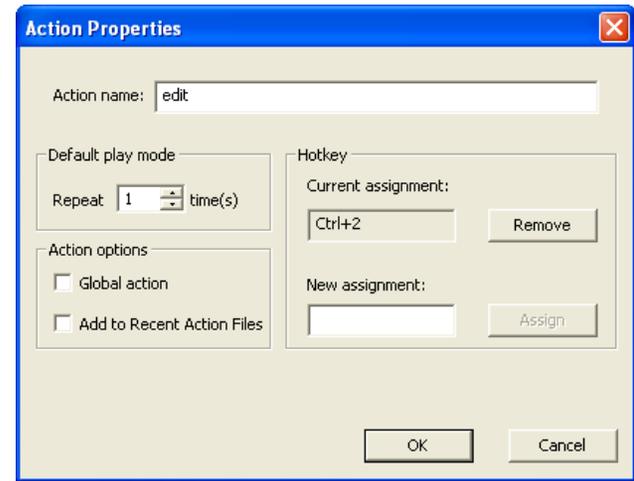
- HDL エディタでは、ウィンドウで操作した一連の手順を記憶させ、再現させることができます。下記の操作を記憶させることができます。
 - メニュー上の HDL エディタに関連した操作
 - ツールボタンでの操作
 - ショートカットキーでの操作
 - ブレークポイント設定操作
 - 基本的なファイル操作
 - ファイル編集操作
 - ナビゲーション操作



アクションレコーダー

操作方法

- Action Properties ウィンドウの表示
 - ツールボタン  をクリック
 - Ctrl + Alt + R
 - HDL エディタ上を右クリック | HDE User Actions Recorder | Start recording
- アクション名を入力
- Default play mode セクションに実行させたい回数を入力
- ウィンドウのOKボタンをクリック。実行した操作は .pl ファイルとして保存されます。
- ストップ
 - ツールボタン  をクリック
 - Ctrl + Alt + S
- ポーズ
 - ツールボタン  をクリック
 - Ctrl + Alt + P



アクションプレイヤー

操作方法

- プレイ
 - ツールボタン  をクリック
 - Alt + M
 - HDL エディタ上を右クリック | HDE User Actions Player | Play Action
 - action コマンドでの操作も可能

