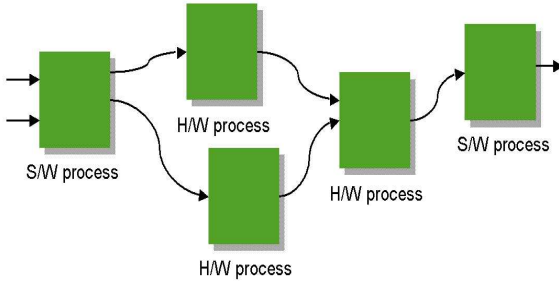


# Impulse C™ ~ CoDeveloper™

「C言語RTL-HDL動作合成」+ PowerPC、MicroBlaze、Nios でhw/sw協調設計も可能

## CSP を元にした わかりやすい並列動作モデリング

\* CSP = Communicating Sequential Process



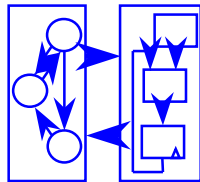
- ✓ 各プロセスは ANSI-C の関数
- ✓ プロセス間結合はConfiguration関数で
- ✓ プロセス単位でハード/ソフトの指定

## シミュレーション機能

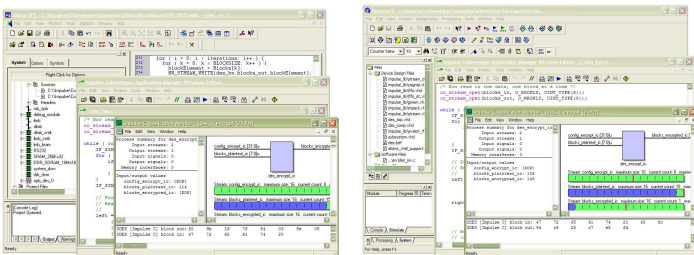
- ✓ プロセス動作、プロセス間のデータの流の観察が可能
- ✓ サイクル・アキュレート・シミュレーションもサポート(05 Q4)

## C言語RTL-HDL動作合成

- ✓ ループのパイプライン化
- ✓ ステージサイズのコントロール
- ✓ 合成の詳細レポート~グラフィカル
- ✓ ステートマシン+データパスに合成
- ✓ 判り易い合成結果
- ✓ VHDL/ Verilog両出力のサポート



Xilinx ISE/EDK  
Altera Quartus /SOPC Builder への  
シームレスな接続



- 前面 ~ CoMonitorシミュレーション
- 背面 ~ それぞれEDK、SOPC Builderのシステム構成タブ(設計IPの自動登録)

(輸入元) InterLink 有限会社 インターリンク

TEL: 045-663-5940 e-mail: ilink\_sales@ilink.co.jp

## 広い活用領域

- 1) Switch 読み込み、LED ドライブ から
- 2) CアルゴリズムモジュールのHDL変換 そして
- 3) CSPモデリングによる高度な協調設計 まで



(例) スパコン Cray XD1もサポート!

## C言語 RTL-HDL 動作合成 ~ VHDL/ Verilog

### FIRフィルタの合成例(一部分)

```
#pragma CO PIPELINE
    accum = 0;
    for (tap = 0; tap < TAPS; tap++) {
#pragma CO UNROLL
#pragma CO SET StageDelay 100

        accum += firbuffer[tap] * coef[tap]; }
    nFiltered = accum >> 2;
    co_stream_write(filter_out, &nFiltered, sizeof(int32));
    IF_SIM(sampleswritten++);

    for (tap = 1; tap < TAPS; tap++) {
#pragma CO UNROLL
        firbuffer[tap-1] = firbuffer[tap];
    }
}
```



```
.....
-- b1 stage 4
s_b1_vstall(4) <= '0';
ni30_accum <= add(s_b1_3_f_accum, mul2_s(r_firbuffer, r_coef)(31 downto 0));
ni33_tap <= X"00000003";
s_b1_vbreak(4) <= '0';
process (clk)
begin
    if (clk'event and clk='1') then
        if (s_b1_vwrite(4) = '1') then
            s_b1_4_f_tap <= ni3693_tap;
            s_b1_4_f_accum <= ni3690_accum;
        end if;
    end if;
end process;

-- b1 stage 5
s_b1_vstall(5) <= '0';
ni32_accum <= add(s_b1_4_f_accum, mul2_s(r_firbuffer, r_coef)(31 downto 0));
.....
```

## OPB/ FSL/ PLB/ APU /Avalon I/F生成

- ✓ ANSI C hw/sw協調設計を実現
- ✓ EDK (XPS) SOPC Builder のプロジェクト生成

Impulse CのモジュールはEDK、SOPC Builderのユーザ設計IPとして自動登録