

ASIAN記法改訂D (2010/5/23)
有限会社 プロエクシィ

1. はじめに

PLDやFPGA等のプログラムを回路図で記述すると、レジスタが視認できるので信号の時間的流れが判り易いのですが、HDL等のテキストによる記述は、信号と時間の関係（波形）が表現し難いので、時間軸に対する動作が判り難いという欠点があります。

回路規模が大きくなりファイルを複数に分割する様になると、その傾向はさらに強まり回路全体の見通しが悪くなります。

ASIAN記法 (Attributed SIgnAl Naming 俗称エイジアン記法)は、信号名の最初に数文字からなる接頭語を付けて波形を表現する様にしたものです。

本記法によると複数のシステムクロックにおける信号の属すクロック系統が一目瞭然になり、非同期回路／同期回路の明瞭化、モノステーブル対策漏れ、ホールド信号のリセット条件漏れ等の基本的な凡ミスもし難くなり、プログラムの信頼性が向上します。

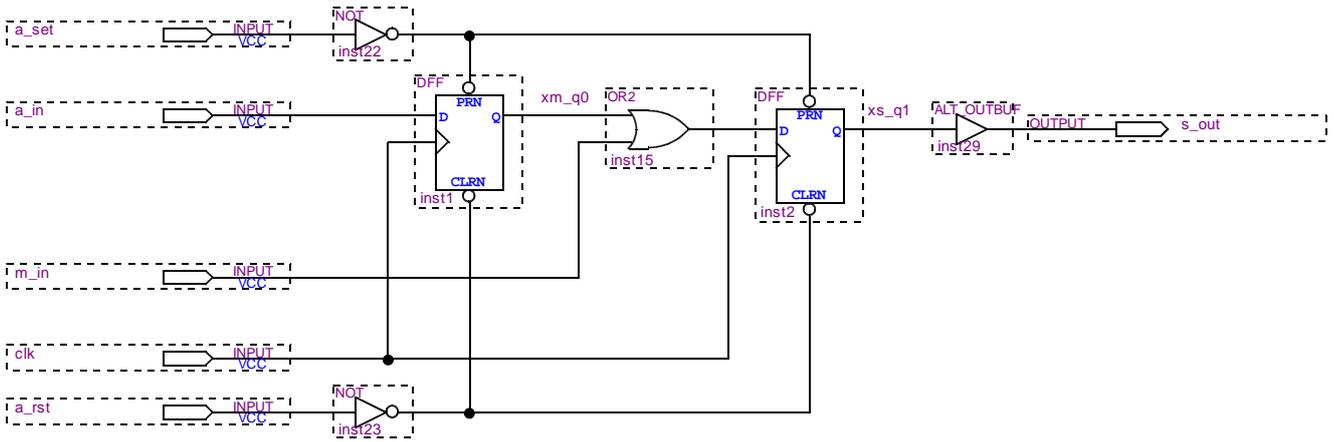
2. ASIAN記法の詳細

信号名の先頭に、各HDLで使用を許された文字で、下記の接頭語分類表に沿った信号属性を示す接頭語を付けます。説明の為に接頭語と回路図の対応を次頁に示します。

接頭語分類表

接 頭 語	説 明
s_	s(ynchronous) システムクロックに同期した信号（レジスタ出力）。
a_	a(synchronous) 非同期信号
c_	c(ombinatorial) レジスタ等でシステムクロックに同期させた信号をAND, OR, XOR等の組み合わせ回路で論理合成した信号。 (次段のレジスタのセットアップタイムを満足しない可能性やハザードを含む可能性がある信号である)
h_	h(oldded) レジスタでホールド（ラッチ）された信号。 (セット信号とリセット信号を要す)
.p_	p(ulsed) 1 システムクロック幅にパルス化された信号。 s, cと組み合わせて sp_, cp_ の様に表わす。 (状態変化検出信号としてラッチイネーブル信号、クロックイネーブル信号等に使用される場合が多い)
f_	f(ixed) 定数、条件設定用スイッチ入力等の、スタティックな信号。
m_	m(etastable) メタステーブル状態になる可能性を含む信号。 (非同期信号をレジスタで最初にラッチした信号)
.r_	システムクロック立ち上がり(rise)に同期を明示する場合に付加
.f_	システムクロック立ち下がり(fall)に同期を明示する場合に付加
.n_	システムクロックが複数ある場合は clk _n に同期した信号は n を付して s1_, s2_, sp1_, c3_ 等の様に表わす。
x.._	内部信号を示す。 (xを外せばそのまま入出力信号名 .._ に編集できる)
.v_	ベクタタイプの信号を示す。

3. A S I A N 記法の使用例



上図で示すDFFをASIAN記法によるVHDLでの記述例を以下に示します。

なお、同回路は非同期信号の a_in、又は外部でclkによりDFFで一回クロッキングされた m_in の何れかを入力とするもので、使用しない他方の信号は外部で L に固定して用います。

```
-----
--      DFF
--      FileName : d_ff.vhd
-----

library ieee;
use ieee.std_logic_1164.all;

entity d_ff is
    port(
        clk          : in std_logic;
        a_rst        : in std_logic;
        a_set        : in std_logic;
        a_in         : in std_logic;
        m_in         : in std_logic;
        s_out        : out std_logic
    );
end d_ff;

architecture rtl of d_ff is
    signal xm_q0      : std_logic;
    signal xs_q1      : std_logic;

begin
    -- shift register and dff
    process (a_rst, clk) begin
        if (a_rst = '1') then
            xm_q0 <= '0';
            xs_q1 <= '0';
        elsif (a_set = '1') then
            xm_q0 <= '1';
            xs_q1 <= '1';
        elsif (clk'event and clk='1') then
            xm_q0 <= a_in;
            xs_q1 <= xm_q0 or m_in;
        end if;
    end process;

    -- set output
    s_out <= xs_q1;

end rtl;
```