

Apresentamos neste documento uma série de modificações realizadas na distribuição original do processador Plasma para a inserção do periférico Perif_exp3, o qual possui propriedade de interrupção. O novo periférico substitui o módulo Ethernet-DMA, existente na distribuição original do Plasma. Caso ambos sejam usados, modificações mais cuidadosas devem ser realizadas.

O documento é apresentado como uma série de procedimentos a serem adotados sobre a distribuição original do Plasma (referenciados no caso do Perif_exp3), porém o estudante poderá realizar mudanças similares sempre que desejar incluir outro novo periférico.

1. Modificações no Plasma.VHD

A)

Na definição da entidade, adicionar os seguintes portos de entrada e saída:

```
ext          : IN STD_LOGIC;
data_in_ext  : IN STD_LOGIC_VECTOR(31 DOWNT0 0);
data_out_ext : OUT STD_LOGIC_VECTOR(31 DOWNT0 0);
```

B)

Nas declarações da arquitetura, antes do begin, adicionar o seguinte:

```
SIGNAL data_perif_exp3_to_mlite : STD_LOGIC_VECTOR(31 DOWNT0 0);
SIGNAL w_e                      : STD_LOGIC;
SIGNAL r_e                      : STD_LOGIC;
SIGNAL irq_perif_exp3           : STD_LOGIC;
SIGNAL enable_perif_exp3        : STD_LOGIC;
SIGNAL enable_perif_exp3_read   : STD_LOGIC;
SIGNAL enable_perif_exp3_write  : STD_LOGIC;
```

```
COMPONENT perif_exp3
```

```
  GENERIC (longbits: natural:=32);
```

```
  PORT( SIGNAL clock      : IN STD_LOGIC;
        SIGNAL reset      : IN STD_LOGIC;
        SIGNAL ext        : IN STD_LOGIC;
        SIGNAL w_e        : IN STD_LOGIC;
        SIGNAL r_e        : IN STD_LOGIC;
        SIGNAL irq        : OUT STD_LOGIC;
        SIGNAL data_in_ext : IN STD_LOGIC_VECTOR(longbits-1 DOWNT0 0);
        SIGNAL data_out_ext : OUT STD_LOGIC_VECTOR(longbits-1 DOWNT0 0);
        SIGNAL data_in_mlite : IN STD_LOGIC_VECTOR(longbits-1 DOWNT0 0);
```

```
SIGNAL data_out_mlite : OUT STD_LOGIC_VECTOR(longbits-1 DOWNT0 0));  
END COMPONENT;
```

C)

Eliminar (deixar como comentário) a definição do IRQ_STATUS (a terceira designação dentro da arquitetura, depois do *begin*):

```
irq_status <= gpioA_in(31) & not gpioA_in(31) & irq_eth_send & irq_eth_rec & counter_reg(18)  
& not counter_reg(18) & not uart_write_busy & uart_data_avail;
```

Adicionar em seu lugar a definição:

```
irq_status <= gpioA_in(31) & not gpioA_in(31) & irq_perif_exp3 & '0' & counter_reg(18) & not  
counter_reg(18) & not uart_write_busy & uart_data_avail;
```

D)

Eliminar a instrução (deixar como comentário)

```
enable_eth <= '1' when enable_misc = '1' and mem_address(7 downto 4) = "0111" else '0';
```

Adicionar em seu lugar a definição:

```
enable_eth <= '0';
```

E)

Adicionar novas instruções da seguinte forma, copiando-as antes do primeiro port map (LABEL: u1_cpu) :

```
enable_perif_exp3 <= '1' when enable_misc = '1' and mem_address(7 downto 4) = "0111" else '0';  
enable_perif_exp3_read <= enable_perif_exp3 and not write_enable;  
enable_perif_exp3_write <= enable_perif_exp3 and write_enable;
```

F)

Na lista de sensibilidade do processo *misc_proc*, adicionar o sinal "data_perif_exp3_to_mlite".

G)

Adicionar no corpo processo *misc_proc*, dentro do case: "case mem_address(6 downto 4) is" a instrução:

```
when "111" =>    --modulo_perif_exp3
    data_r <= data_perif_exp3_to_mlite;
```

H)

Adicionar a seguinte instanciação na arquitetura ao final do arquivo:

```
modulo: perif_exp3
    GENERIC MAP (longbits => 32)
        PORT MAP(
            clock          => clk,
            reset          => reset,
            ext            => ext,
            w_e           => enable_perif_exp3_write,
            r_e           => enable_perif_exp3_read,
            irq           => irq_perif_exp3,
            data_in_ext   => data_in_ext,
            data_out_ext  => data_out_ext,
            data_in_mlite => data_w,
            data_out_mlite => data_perif_exp3_to_mlite);
```

2. Modificações no Plasma_tbw.VHD

ATENÇÃO: NÚMEROS ESPECÍFICOS ASSOCIADOS AO NÚMERO USP

Você deverá gerar dois números que serão utilizados no item D) abaixo:

primeiro_num = composto pelos primeiros quatro algarismos do seu número USP

segundo_num = composto pelos últimos quatro algarismos do seu número USP

Exemplo: número USP= 7350422

primeiro_num= 7350

segundo_num= 0422

A)

Na definição do componente Plasma, introduzir na mesma sequência da definição da entidade Plasma (veja item 1.A) :

```
ext          : IN STD_LOGIC;
data_in_ext  : IN STD_LOGIC_VECTOR(31 DOWNT0 0);
data_out_ext : OUT STD_LOGIC_VECTOR(31 DOWNT0 0);
```

B)

Adicionar os seguinte sinais:

```
SIGNAL ext          : STD_LOGIC :='0';
SIGNAL data_in_ext  : STD_LOGIC_VECTOR(31 DOWNTO 0) :=(OTHERS=>'0');
SIGNAL data_out_ext : STD_LOGIC_VECTOR(31 DOWNTO 0);
```

C)

Na instanciação do Plasma, adicionar:

```
ext          => ext,
data_in_ext  => data_in_ext,
data_out_ext => data_out_ext,
```

D)

Dentro do processo tb, após as linhas

```
reset <= '1';
wait for 2*PERIOD;
```

```
reset <= '0';
wait for PERIOD;
```

Adicionar as seguintes linhas:

```
wait for 1500*PERIOD;
```

```
data_in_ext <= std_logic_vector(to_unsigned (XXXX, 32)); /* usar o primeiro_num
wait for 2*PERIOD;
```

```
ext <= '1';
wait for 1*PERIOD;
```

```
ext <= '0';
wait for 1*PERIOD;
```

```
wait for 500*PERIOD;
```

```
data_in_ext <= std_logic_vector(to_unsigned (XXXX, 32)); /* usar o segundo_num
wait for 2*PERIOD;
```

```
ext <= '1';
wait for 1*PERIOD;
```

```
ext <= '0';  
wait for 1*PERIOD;
```

E)

No topo do arquivo, adicionar a biblioteca (para que a operação de transferência de *integer* para *std_logic_vector* possa ser feita na designação de *data_in_ext*):

```
use ieee.numeric_std.all;
```