1. Introduction

CME-HR devices have 3 configuration modes: JTAG, AS and PS mode. In the PS mode, CME-HR family works as a slave device, receives configuration data from external master controller passively. SPI Master cannot read configuration data from CME-HR family.

2. Configuration Mode and Pins

The bitstream is loaded into the device through special configuration pins. These configuration pins serve as the interface for a number of different configuration modes:

There are 3 configuration modes: JTAG, AS and PS mode which are defined by two dedicated JM_B and SS pins. After exiting the Power-On Reset (POR) state or when nCONFIG returns High after being held Low, the CME-HR device samples the logical value on its SS and JM_B pins.

The configuration modes are described in table below.

Table 1 Configuration Mode

Mode pin		Mode	Description
JM_B	SS		
X	1	AS	Active Serial mode. The chip will be configured automatically. Configuration data is stored in the SPI flash.
X	0	PS	Chip acts as slave. External microcontroller feeds configuration data into the chip.
0	X	JTAG	JTAG-based configuration. This mode takes high precedence over AS and PS modes.
1	X	AS/PS	JTAG can't configure the CME-HR.

Table 2 Configuration Pins

SPI serial configuration Pins		
SCLK	Input/output	In passive serial configuration mode, SCLK is a clock input used to clock configuration data from external device source into device. In active serial configuration mode, SCLK is a clock output from device. The pin can be used as regular user I/Os after configuration.
SDI	output	Dedicated configuration data output pin in AS mode. No configuration function in PS mode. The pin can be used as regular user I/Os after configuration in AS mode
SDO	Input	Serial data input from external master in PS mode or from spi flash in AS mode.

SPI serial configuration Pins		
		The pin can be used as regular user I/Os after configuration.
SS	output or input	Chip select output to enable a SPI Flash in AS mode or input as a CME device select. This output is used during AS mode. The pin can be used as regular user I/Os after configuration in AS mode. it is used as chip selection control (input) during PS.The pin can be used as regular user I/Os after configuration in PS mode.
Dedicated Configuration Pins:		
CFGDONE	output	This is a dedicated configuration status pin, the pin will output high during configuration. The pin can be used as regular user I/Os after configuration.
nCONFIG	input	Chip global reset input. Active low.

3. PS Scheme

The PS schematic is shown below. The CME-HR's SPI configuration interface is a separate, independent I/O bank, powered by the VDD2V5 supply input. Typically, VDD2V5 is the same voltage as the application processor's I/O. The configuration control signals, CFGDONE and nCONFIG, are supplied by the separate I/O Bank 1 voltage input, VDDIO_1.

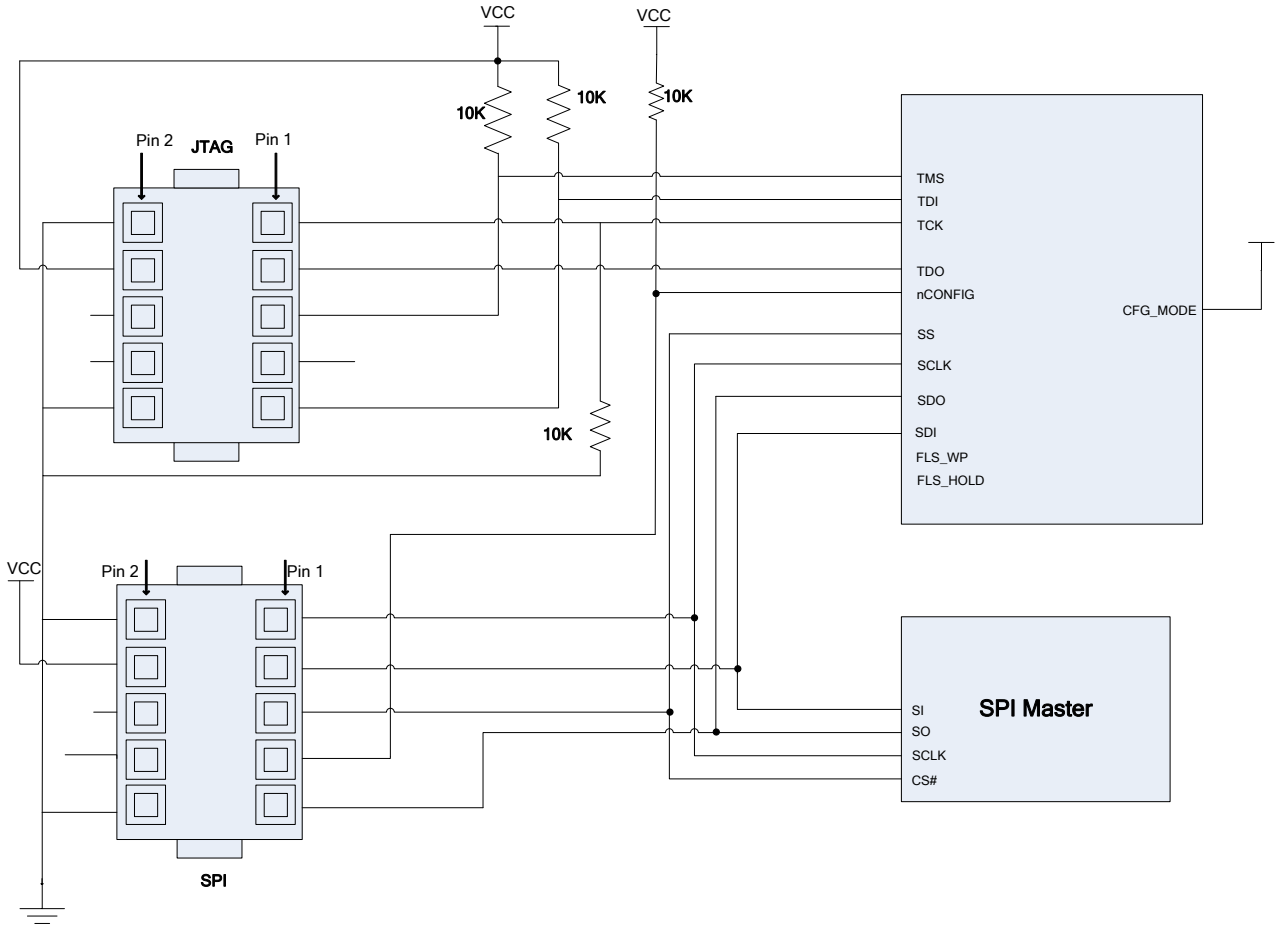

Figure 1 PS schematic

4. PS Configuration Process

You can perform PS configuration on CME-HR device family with an external intelligent host, such as a microprocessor with flash memory. In the PS scheme, an external host controls the configuration process. After configuration, the SPI port pins are available to the user-application as additional PIO pins, supplied by the VDD2V5 input voltage.

Figure 2 illustrates the interface timing for the PS mode.

1) Reset process. The application host begins by driving the CME-HR nCONFIG pin low, resetting the CME-HR FPGA. Similarly, the host holds the CME-HR's SS pin Low. The host must hold the nCONFIG pin Low for at least 500 ns. Ultimately, the host either releases the nCONFIG pin and allows it to float High via the 10 KOhm pull-up resistor to VCCIO_1 or drives nCONFIG High. The CME-HR FPGA enters PS mode when the nCONFIG pin returns High while the SS pin is Low

2) Initialization process. After driving nCONFIG High or allowing it to float High, the host must wait a minimum of 1250 µs, allowing the CME-HR device clears the internal configuration memory.

3) Clock switch process. After the initialization process, the host must drive the SS to high and the high time is 8 SCLK cycles.

4) Configuring internal configuration memory process. After the configuration clock is switched to SCLK, the host sends the configuration image(*_ps.acf) generated by the Primace. The host must place the configuration data one bit at a time on the SDI pin at the negedge of the SCLK. The most-significant bit(MSB) of each byte must be sent first. For example, if the configuration data *_ps.acf contains the byte sequence 02 1B EE 01 FA, the serial bitstream you must send to the device is:

   0000-0010 0001-1011 1110-1110 0000-0001 1111-1010

   CME-HR device family receives configuration data on the SDI pin and the clock is received on the SCLK pin. Data is latched into the device on the rising edge of SCLK. Data is continuously clocked into the target device until all the configuration data is sent and the SS is driven to high.

5) Addition dummy process. After sending the entire image, the CME-HR FPGA CONF_DONE goes high and the device enters the user mode. If the CONF_DONE pin remains Low, then an error occurred during configuration and the host should handle the error accordingly for the application. The host must send extra 64 SCLK clock cycles measured from rising-edge to rising-edge to CME-HR device. After the additional SCLK cycles, the SPI interface pins then become available to the user application loaded in FPGA.

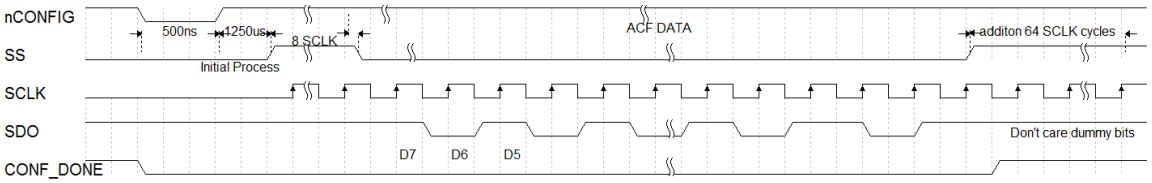

Figure 2 PS Configuration Waveform

To reconfigure the CME-HR FPGA or to load a different configuration image, merely restart the configuration process by pulsing nCONFIG Low or power-cycling the FPGA.

The max PS SCLK clock frequency is about 25MHz.

5. PS Configuration Reference Code

The PS code example is shown below. The 8051 MCU PS host uses the GPIO to control the PS master's signals. The below code signals are from the master's point of view. The LBA_Buff array

```
#define CS P0_0 //CS
#define SDO P0_1 //SDO
#define SCLK P0_2 //SCLK
#define nCONFIG P0_3 // nCONFIG
#define FILE_LEN        0xAB000

unsigned char HR3_PS_pre_clk;
unsigned int i;
unsigned char i_mask;
unsigned char bit_mask;

nCONFIG = 1;
CS=0;
SDO =1;
SCLK =1;
//Reset process
CS=0;
nCONFIG = 0;
Delay1us();
nCONFIG = 1;
```

```
//Initialization process
Delay1250us();

// Clock switch process
CS=1;
for(HR3_PS_pre_clk=0;HR3_PS_pre_clk<8;HR3_PS_pre_clk++) //HR3 8=pass
{
        SCLK=0;//clk
        Delay1us();
        SCLK=1;//clk
        Delay1us();
 }
 CS=0;    //cs
// Configuring internal configuration memory process
for(i=0;i< FILE_LEN /256;i ++)
        GetNextBlockData();    //Get next 256 bytes data to LBA_Buff buffer.
        for(ps_x=0;ps_x<256;ps_x++)
         {
                bit_mask=0x80;
                for(i_mask=0;i_mask<8;i_mask++)
                {
                        SCLK=0;//clk
                        Delay1us();
                        if(LBA_Buff[ps_x] & bit_mask)//data
                                SDO=1;
                        else
                                SDO=0;
                        Delay1us();
                        SCLK=1;//clk
                        Delay1us();
                        bit_mask>>=1;
                 }
```

```
                Delay1us();
            }
}
CS=1;
 // Addition dummy process
   for(i_mask=0;i_mask<64;i_mask++)
   {
            SCLK=0;//clk
            Delay1us();
            SCLK=1;//clk
             Delay1us();
   }
   SCLK=1;//clk
```