

开源 H.265 Encoder IP 使用指南

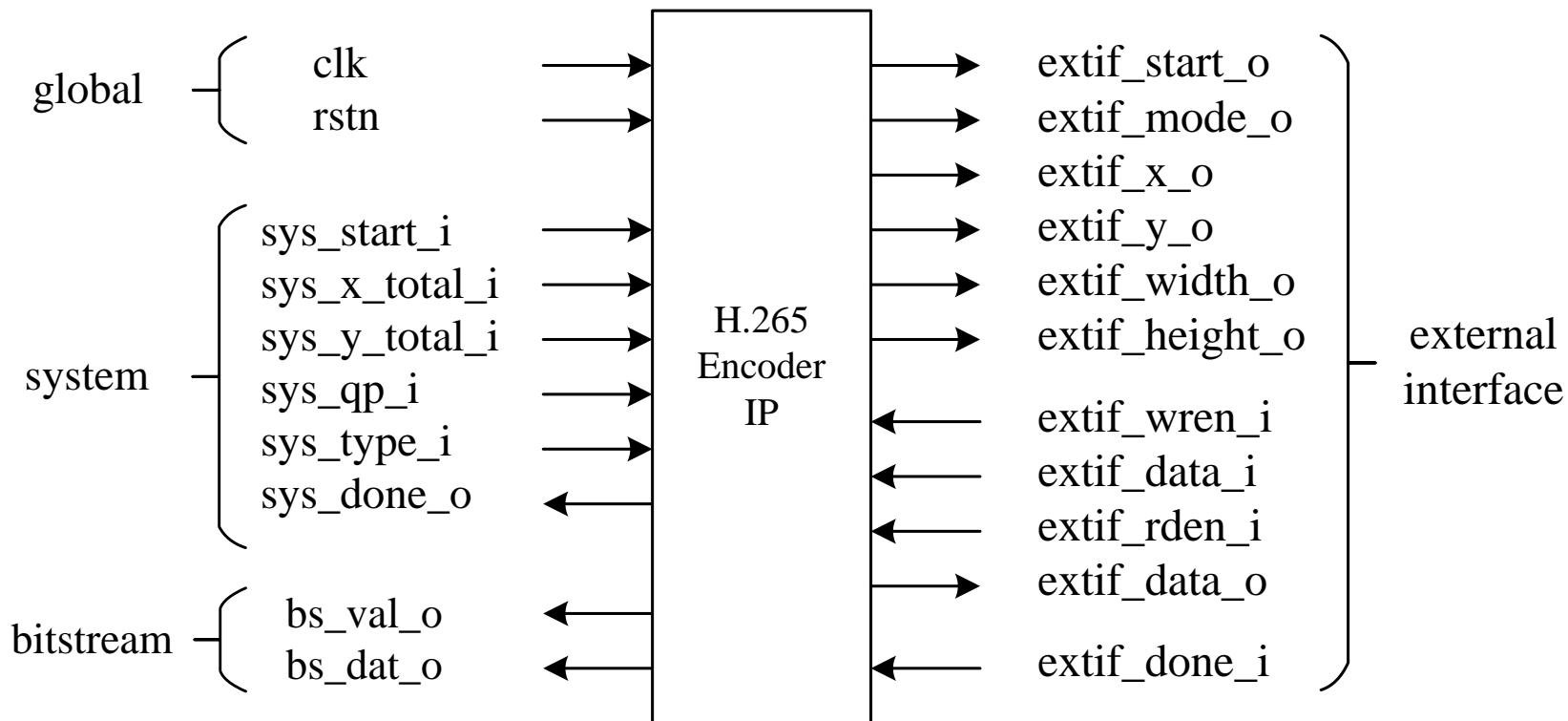
作者 NoBody

日期 2017-01-09

目录

1. 接口介绍
2. 封装示例
3. 硬件系统
4. 软件系统
5. 演示系统

接口介绍 - 总览



接口介绍 – external interface

extif_start_o
extif_mode_o
extif_x_o
extif_y_o
extif_width_o
extif_height_o

extif_wren_i
extif_data_i
extif_rden_i
extif_data_o
extif_done_i

external
interface

parameter	LOAD_CUR_LUMA	=	01	,
	LOAD_REF_LUMA	=	02	,
	LOAD_CUR_CHROMA	=	03	,
	LOAD_REF_CHROMA	=	04	,
	LOAD_DB_LUMA	=	05	,
	LOAD_DB_CHROMA	=	06	,
	STORE_DB_LUMA	=	07	,
	STORE_DB_CHROMA	=	08	;

接口介绍 – external interface

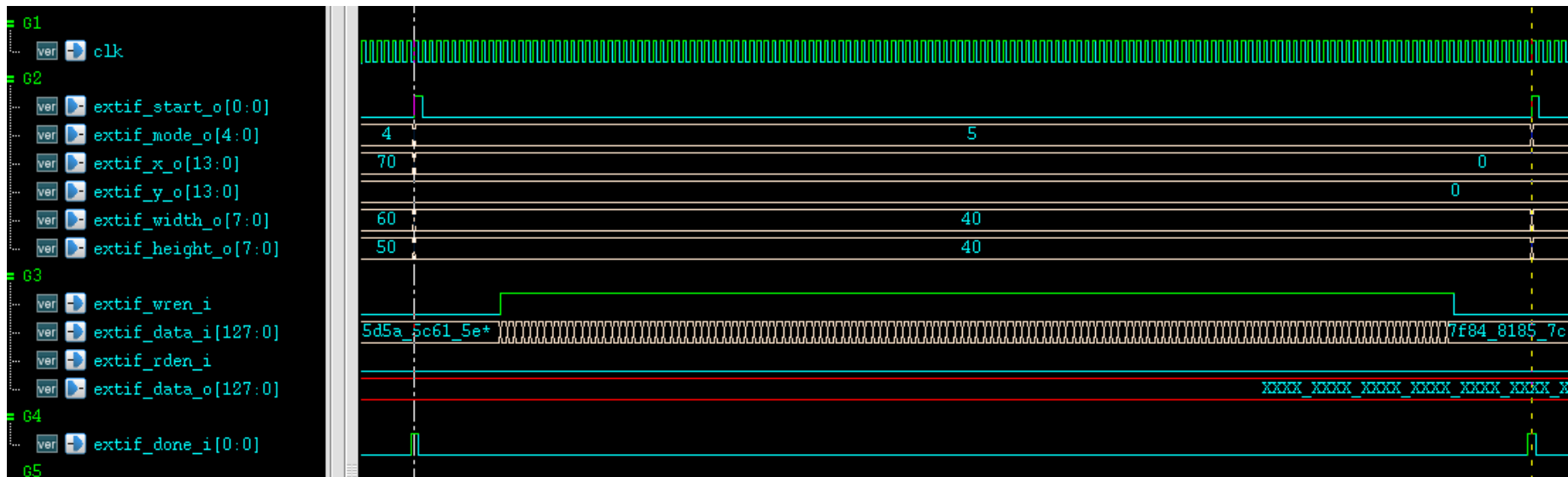
x和y: 给出需要被写入或读出矩形的左上角坐标
width和height: 给出需要被写入或读出矩形的长和宽

wren和rden: 读和写使能

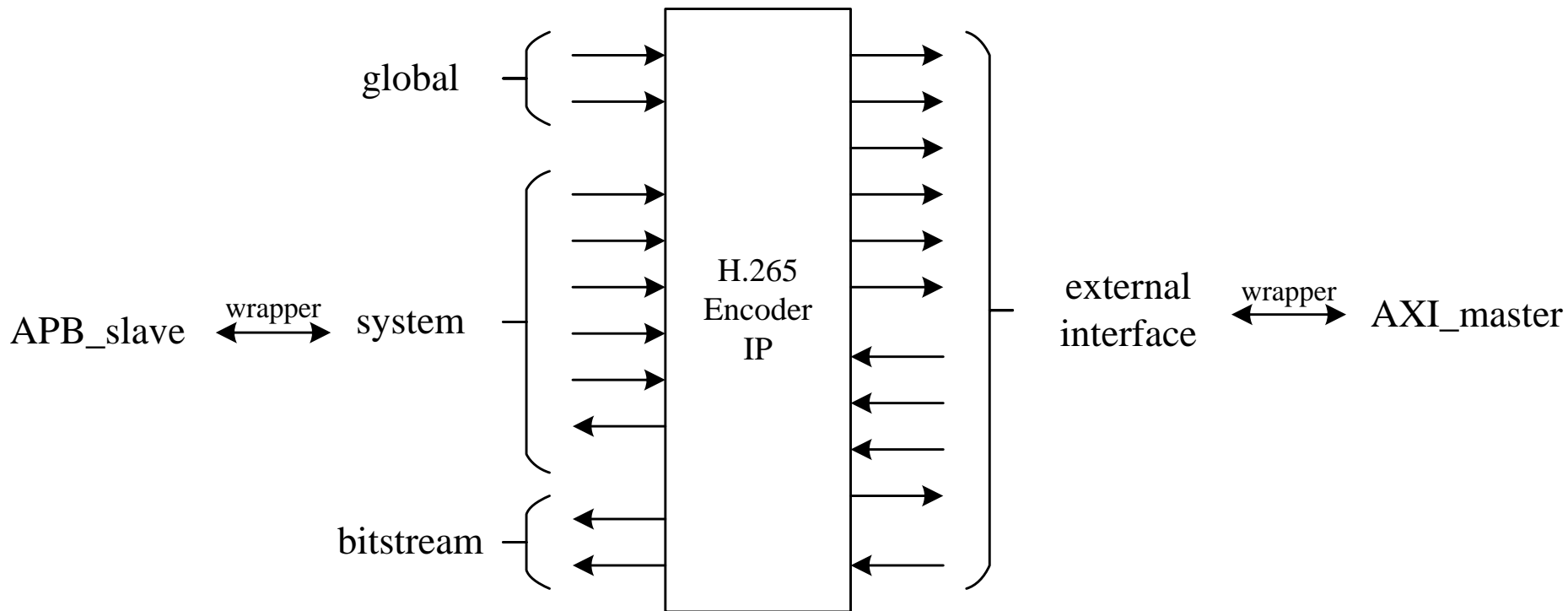
时序与普通sram一致

Luma: yyyyyyyy...

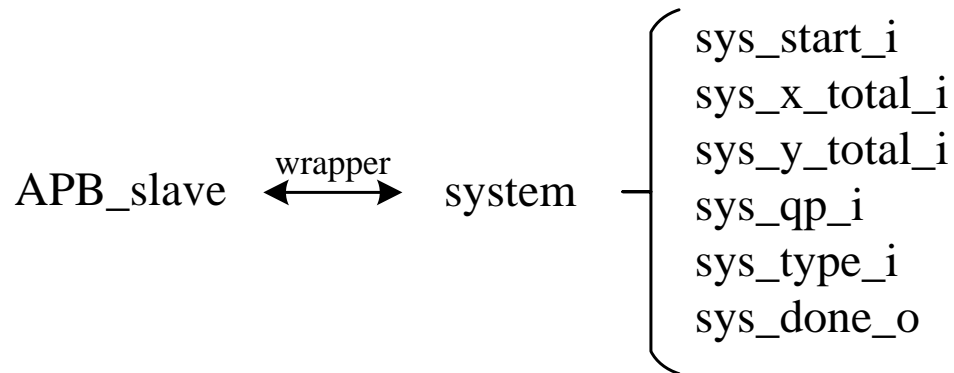
Chroma: uvuvuvuv... 光栅顺序



封装示例



封装示例 - APB_slave



```
parameter ADDR_START = 00 ,  
          ADDR_X_TOTAL = 01 ,  
          ADDR_Y_TOTAL = 02 ,  
          ADDR_MODE = 03 ,  
          ADDR_QP = 04 ,  
          ADDR_TYPE = 05 ,  
          ADDR_ORI_BASE = 06 ,  
          ADDR_REC_0_BASE = 07 ,  
          ADDR_REC_1_BASE = 08 ;
```

封装示例 - AXI_master

```
always_@(*)_begin
    gen_m_mlen = 0 ;
    case( extif_mode_i )
        LOAD_CUR_SUB : gen_m_mlen = 0 ;
        LOAD_REF_SUB : gen_m_mlen = 0 ;
        LOAD_CUR_LUMA : gen_m_mlen = 3 ; // 64 pixels
        LOAD_REF_LUMA : if( addr_offset_x!=0 )
            gen_m_mlen = 5-addr_offset_x/16 ;
            else_if( ((4096-{1'b0, gen_m_maddr[11:0]})<96)&((4096-{1'b0, gen_m_maddr[11:0]})/16)-1_ ;
            else_begin
                gen_m_mlen = 5 ;
            end // 96 pixels
        LOAD_CUR_CHROMA : gen_m_mlen = 3 ; // 64 pixels
        LOAD_REF_CHROMA : if( addr_offset_x!=0 )
            gen_m_mlen = 5-addr_offset_x/16 ;
            else_if( ((4096-{1'b0, gen_m_maddr[11:0]})<96)&((4096-{1'b0, gen_m_maddr[11:0]})/16)-1_ ;
            else_begin
                gen_m_mlen = 5 ;
            end // 96 pixels
        LOAD_DB_LUMA : gen_m_mlen = 3 ; // 64 pixels
        LOAD_DB_CHROMA : gen_m_mlen = 3 ; // 64 pixels
        STORE_DB_LUMA : gen_m_mlen = 3 ; // 64 pixels
        STORE_DB_CHROMA : gen_m_mlen = 3 ; // 64 pixels
    endcase
end
```

data storage format
in external memory
+
data width of axi bus

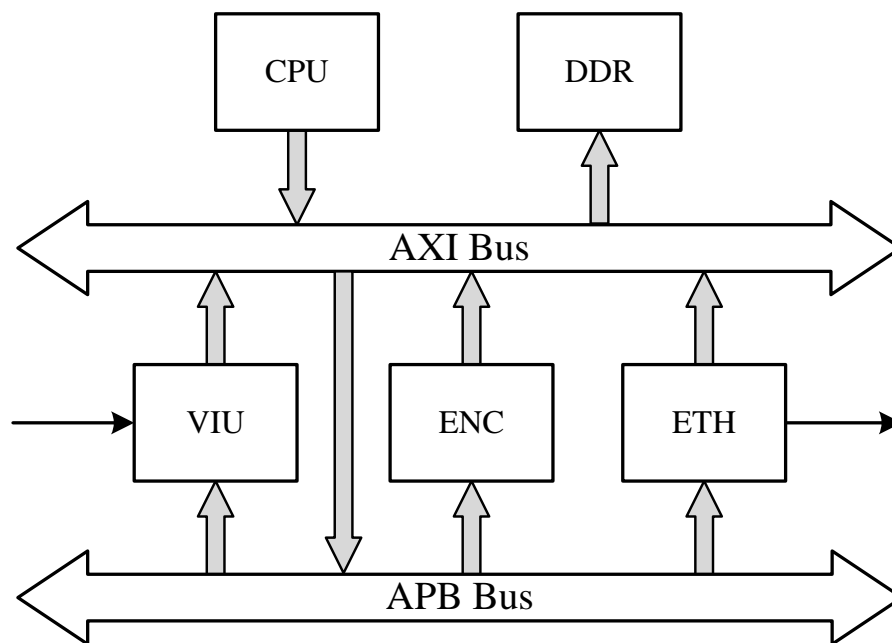
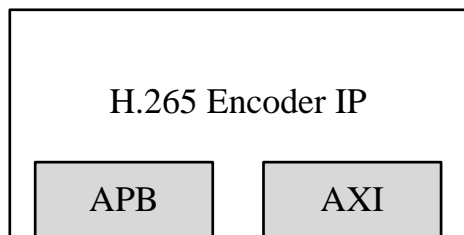


burst length



efficiency

硬件系统

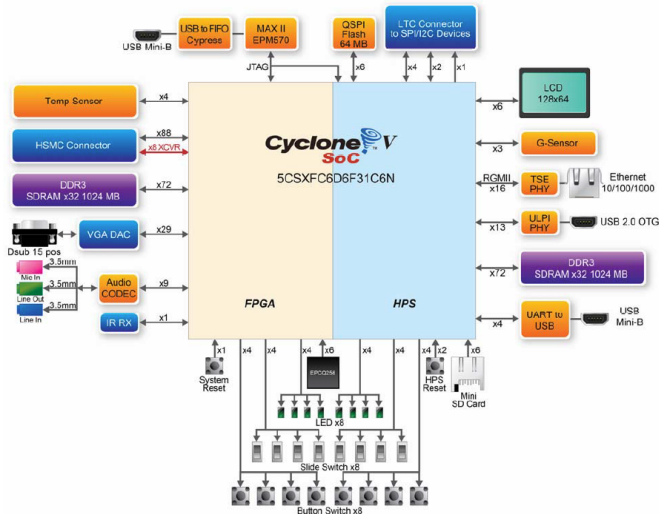


硬件系统 - CPU & ETH

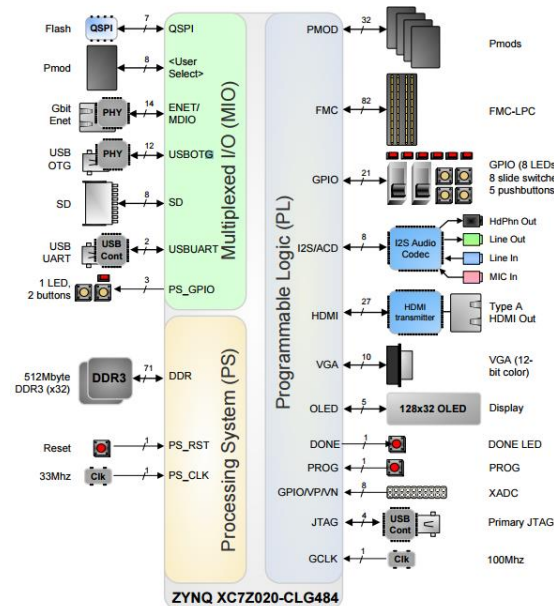
ASIC^o

专注开源硬件 IP Core
www.openasic.org

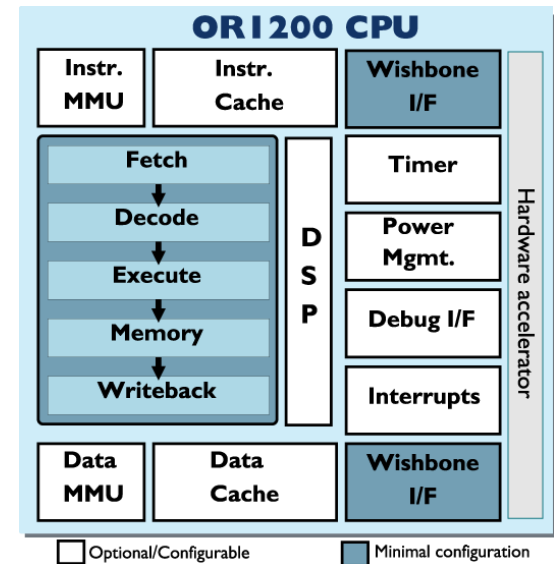
socket



zed



or1200



<http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=205&No=816&PartNo=2>

http://opencores.org/or1k/File:Or1200_blocks.png

http://zedboard.org/sites/default/files/documentations/ZedBoard_HW_UG_v2_2.pdf

硬件系统 - DDR & BUS

Qsys

Use	Connections	Name	Description	Export	Clock	Base	End	
<input checked="" type="checkbox"/>		<input type="checkbox"/> merlin_axi_translator_0	AXI Translator					
		clk	Clock Input	<i>Double-click to export</i>	clk_0			
		clk_reset	Reset Input	<i>Double-click to export</i>	[clk]			
		s0	AXI Slave		[clk]			
		m0	AXI Master		[clk]			
					axi_s			
					<i>Double-click to export</i>			
<input checked="" type="checkbox"/>			<input type="checkbox"/> mem_if_ddr3_emif_0	DDR3 SDRAM Controller with UniPHY				
			pll_ref_clk	Clock Input	clk_pll_ref	exported		
			global_reset	Reset Input	rst_global			
			soft_reset	Reset Input	rst_soft			
			afi_clk	Clock Output	<i>Double-click to export</i>	clk_afi		
			afi_half_clk	Clock Output	<i>Double-click to export</i>	clk_afi_half		
			afi_reset	Reset Output	<i>Double-click to export</i>			
			afi_reset_export	Reset Output	<i>Double-click to export</i>			
		memory	Conduit					
		avl	Avalon Memory Mapped Slave	ddr	clk_afi	# 0x2000_0000	0x3fff_ffff	
		status	Conduit	<i>Double-click to export</i>				
		oct	Conduit	ddr3_status				
		pll_sharing	Conduit	oct				
		dll_sharing	Conduit	pll_sharing				
				dll_sharing				
<input checked="" type="checkbox"/>		<input type="checkbox"/> clk_0	Clock Source					
		clk_in	Clock Input	clk				
		clk_in_reset	Reset Input	rst				
		clk	Clock Output	<i>Double-click to export</i>	clk_0			
		clk_reset	Reset Output	<i>Double-click to export</i>				

硬件系统 - VIU

ASIC[®]

专注开源硬件 IP Core
www.openasic.org

d5m



<http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=68&No=281>

dvi



<http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=68&No=359>

补充 – memory替换

ROM

```
always @(posedge clk) begin
  case( rd_adr )
    0000: rd_dat <= 0000 ;
    0001: rd_dat <= 0001 ;
    0002: rd_dat <= 0002 ;
    0003: rd_dat <= 0003 ;
    0004: rd_dat <= 0004 ;
    0005: rd_dat <= 0005 ;
    0006: rd_dat <= 0006 ;
    0007: rd_dat <= 0007 ;
  endcase
end
```

-> 工具自动识别

```
$readmemb( "rom.dat" , dut.rom );
```

-> 替换成IP，并指定初始化文件

RAM

无需初始化内容 -> 工具自动识别

需要初始化内容 -> 替换成IP，并指定初始化文件

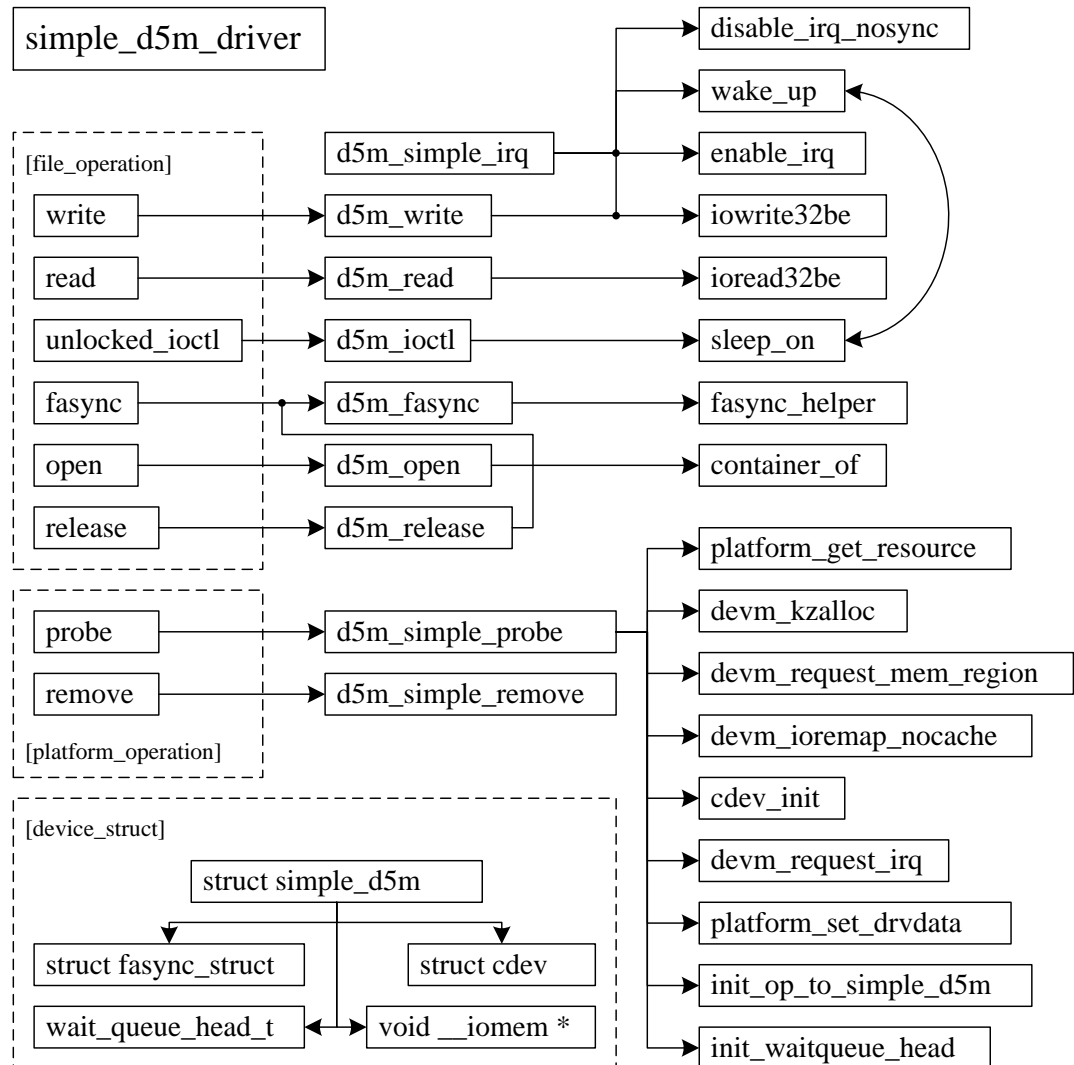
软件系统 - 总览

1. 硬件驱动：初始化和控制运行；
2. NAL打包：本H.265 Encoder输出的码流不包含VPS、SPS、PPS；
3. RTP打包：通过以太网传输。

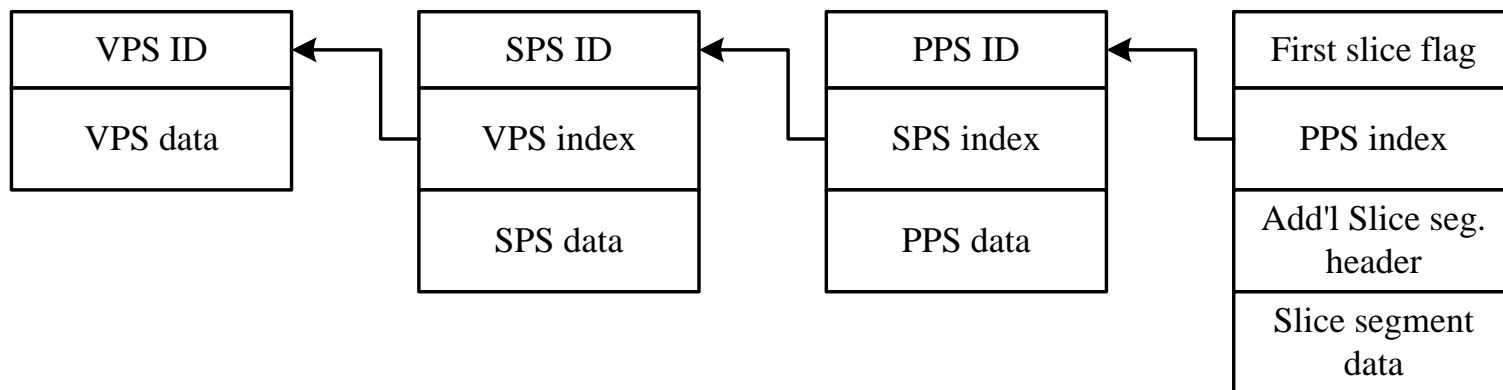
软件系统 - 驱动

```
int_open(const_char *pathname  
         , int_flag  
         , int_perms);
```

```
void* mmap(void* start  
           , size_t_length  
           , int_prot  
           , int_flags  
           , int_fd  
           , off_t_offset);
```



软件系统 - NAL



软件生成的NAL信息



硬件处理得到BS信息

软件系统 - NAL

```
vpsWrite(&bs, _param);  
rtpHeader->Timestamp = htonl(tsCurrent);  
tsCurrent += tsIncrease;  
rtpHeader->SN = htons(seqNum++);  
rtpHeader->M = 1;  
rtpByteCnt = bs.byteCnt + 12;  
send(_socket1, &sendBuf[1], rtpByteCnt, 0);
```

```
#ifdef DEBUG
```

```
dump(&sendBuf[1], 1
```

```
#endif
```

```
SLEEP(SLEEPMILLISEC
```

```
spsWrite(&sliceHeader.sps, &bs, _param);
```

```
rtpHeader->Timestamp = htonl(tsCurrent);
```

```
tsCurrent += tsIncrease;
```

```
rtpHeader->SN = htons(seqNum++);
```

```
rtpHeader->M = 1;
```

```
rtpByteCnt = bs.byteCnt + 12;
```

```
send(_socket1, &sendBuf[1], rtpByteCnt, 0);
```

```
#ifdef DEBUG
```

```
dump(&sendBuf[1], r
```

```
#endif
```

```
SLEEP(SLEEPMILLISEC
```

```
ppsWrite(&sliceHeader.pps, &bs, _param);
```

```
rtpHeader->Timestamp = htonl(tsCurrent);
```

```
tsCurrent += tsIncrease;
```

```
rtpHeader->SN = htons(seqNum++);
```

```
rtpHeader->M = 1;
```

```
rtpByteCnt = bs.byteCnt + 12;
```

```
send(_socket1, &sendBuf[1], rtpByteCnt, 0);
```

```
#ifdef DEBUG
```

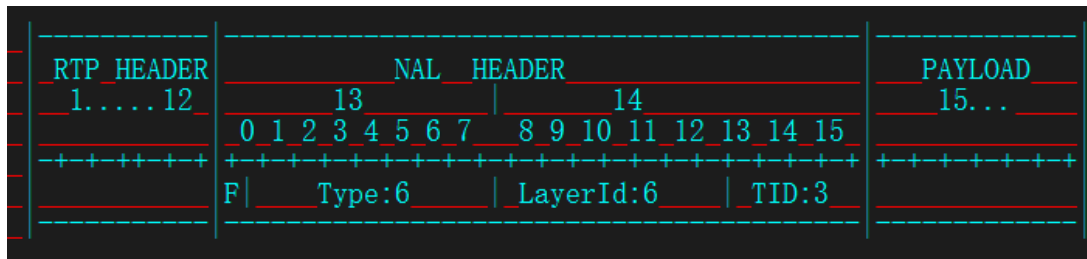
```
dump(&sendBuf[1], rtpByteCnt);
```

```
#endif
```

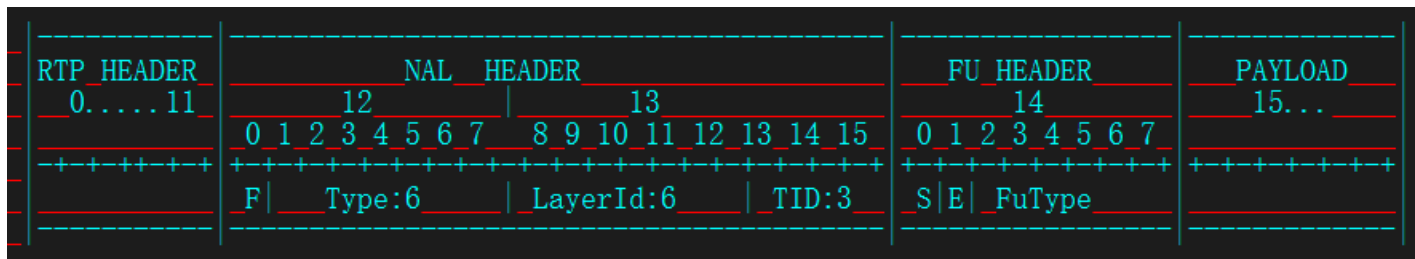
```
SLEEP(SLEEPMILLISECOND);
```

软件系统 - RTP

一次传输



分批传输



演示系统 - 总览

debug mode

UART -> DDR / VIU / ENC

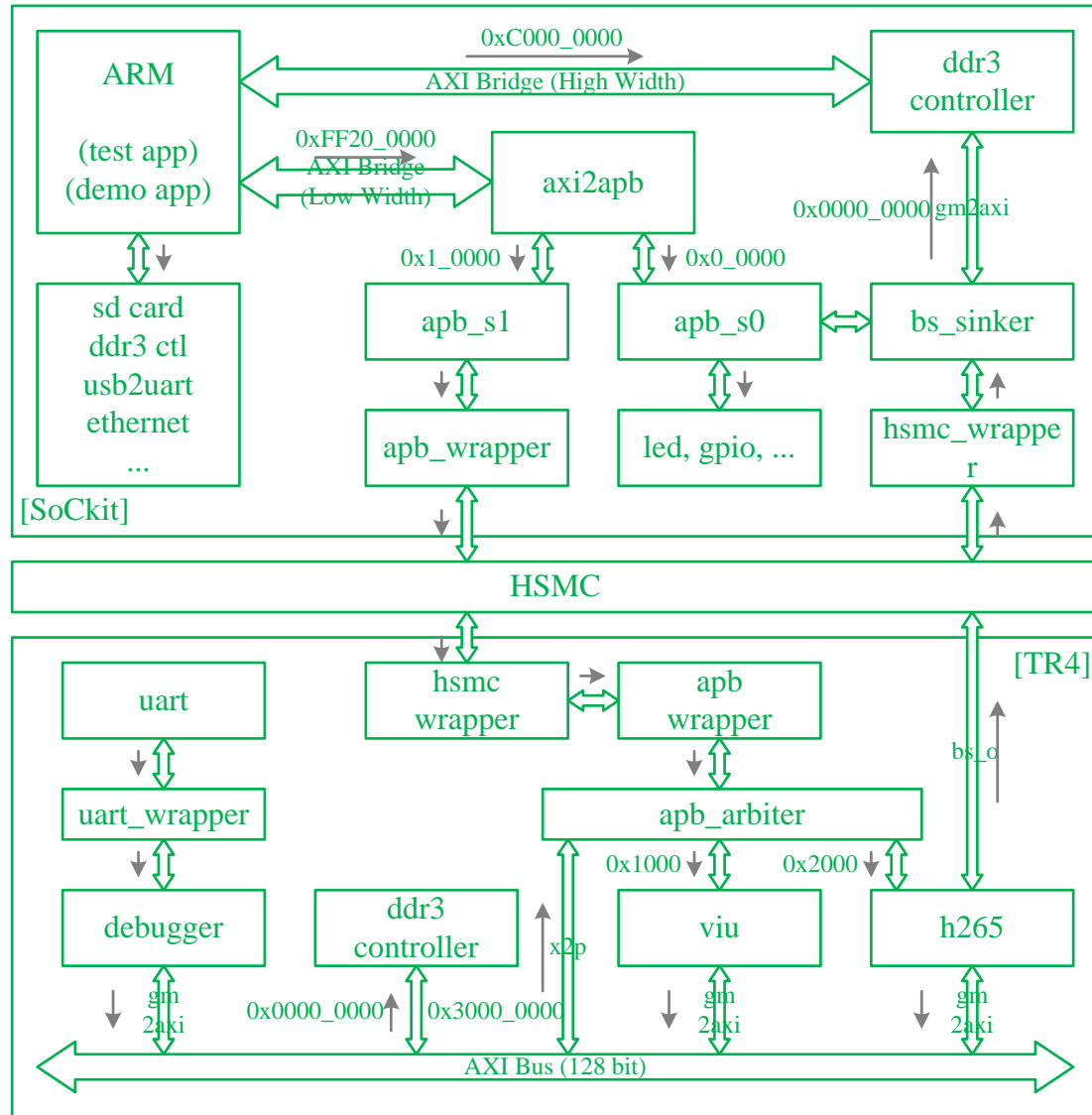
test mode

ARM (test app) -> DDR / ETH

demo mode

ARM (demo app) -> VIU / ENC / ETH

- verified
- simulated
- rtl ready
- rtl not ready
- unknown



演示系统 - 视频

<https://v.qq.com/x/page/f03609q5n16.html>

ASIC^o

专注开源硬件 IP Core

www.openasic.org

谢谢！