# VDEC\_HS TEST REQUEST DOCUMENT

## 1.1 Overview



Figure ‑ Vdec\_hs Block Diagram

**本文只涉及V3 vdec\_hs（对应V2 vdec1）内部节点log打印和平台对比需求。**

Vdec\_hs code block size最大29，即HSSCCH\_PART2 13 bits info加上16 bits CRC。因此，vdec\_hs不再有分窗，每次先计算完forward，再从0状态进行traceback，即可得到解码比特。解码完成后，HW还对数据进行CRC校验和SER计算的处理。**Figure 0‑1中彩色标注的点只代表处理流程中不同的阶段，每个点可能需要打印一到两种数据文件，每个点和测试对比点没有一一对应关系，可合并对比。**

Table 1 解码场景

|  |  |  |
| --- | --- | --- |
| TYPE | CODEBLK\_SIZE | CRC |
| HSSCCH\_PART1 | 8 | N |
| HSSCCH\_PART2 | 29 | Y |
| AGCH | 22 | Y |

## 1.2 Log打印需求

下文按Figure 0‑1中彩色圆中数字序号，标志为“圆序号n”，以此序号说明打印log。

**圆序号1**

节点说明：

V3平台待解码软比特存储在HSPA\_CRAM中。V3每个载波支持的HSSCCH/NS\_HSSCCH/AGCH的数量，软比特存放位置，参考WCDMA V3 HSSCCH\_AGCH Design Specification.doc。

**圆序号2**

节点说明：

Deratematching后的软比特信息

打印要求

三种类型的解码任务都需要打印

打印Label：

XXXX\_DERM

DataLength xx BitWidth 6

打印格式：

每行8个软比特，以tab间隔，16进制打印

打印示例

0x01 0x02 0x03 0x04 0xF1 0xF2 0xF3 0xF4

**圆序号3**

节点说明：

UE specific unmasking后的软比特信息。25.212 4.6A.1 Figure 19A中是先进行ratematching，再进行UE specific masking，其过程是把16 bit UE\_ID先进行 1/2编码，得到（16+8）\*2=48比特mask值，再按HSSCCH\_PART1 ratematching pattern进行puncture，剩下40比特和编码并puncture后的信息比特进行异或。V3硬件实现是在Deratemtaching后的数据流中进行unmask，即这里打印的应该是包括puncture的软比特值。HSSCCH\_PART1时，使用UE\_ID进行unmask，HSSCCH\_PART2和AGCH时使用0x0000进行unmask，即数据不变化。

打印要求

HSSCCH\_PART1：打印unmask后的48个软比特

HSSCCH\_PART2：打印和圆序号2一样的deratematching后软比特

AGCH：打印和圆序号2一样的deratematching后软比特

打印Label：

XXXX\_UNMASK

DataLength xx BitWidth 6

打印格式：

每行8个软比特，以tab间隔，16进制打印

打印示例

0x01 0x02 0x03 0x04 0xF1 0xF2 0xF3 0xF4

**圆序号4**

节点说明：

Forward state metric

打印要求

HSSCCH\_PART1：共打印（8 info + 8tail）\*256个state metric

HSSCCH\_PART2：共打印（29 info + 8tail）\*256个state metric

AGCH：共打印（22 info + 8tail）\*256个state metric

打印Label：

XXXX\_STATE\_METRIC

DataLength xx BitWidth 8

打印格式：

每行8个state metric，以tab间隔，16进制打印

打印示例

0x01 0x02 0x03 0x04 0xF1 0xF2 0xF3 0xF4

**圆序号5**

节点说明：

Backword survive path

打印要求

HSSCCH\_PART1：共打印（8 info + 8tail）\*8\*32b

HSSCCH\_PART2：共打印（29 info + 8tail）\*8\*32b

AGCH：共打印（22 info + 8tail）\*8\*32b

打印Label：

XXXX\_SURVIVE\_PATH

DataLength xx BitWidth 32

打印格式：

每行8个32 bit survive path information，以tab间隔，16进制打印

打印示例

0x00000001 0x00000002 0x000000003 0x00000004 0xFFFFFFF1 0xFFFFFFF2 0xFFFFFFF3 0xFFFFFFF4\n

**圆序号6**

节点说明：

Trace back state sequence

打印要求

HSSCCH\_PART1：共打印（8 info + 8tail）\*8b

HSSCCH\_PART2：共打印（29 info + 8tail）\*8b

AGCH：共打印（22 info + 8tail）\*8b

打印Label：

XXXX\_TRACE\_STATE

DataLength xx BitWidth 8

打印格式：

每行8个8 bit traceback state，以tab间隔，16进制打印

打印示例

0x01 0x02 0x03 0x04 0xF1 0xF2 0xF3 0xF4

**圆序号7**

节点说明：

Decoded bits

打印要求

HSSCCH\_PART1：共打印8b

HSSCCH\_PART2：共打印29b

AGCH：共打印22b

打印Label：

XXXX\_DEC\_BITS

DataLength 1 BitWidth 32

打印格式：

只有一个32bit 16进制数，**先接收到的bit在LSB，后接收到的比特在MSB，高位补0**

打印示例

0x00000089

**圆序号8**

节点说明：

Unmasked CRC value和CRC校验结果

打印要求

HSSCCH\_PART1：没有CRC，所以**不打印此LABEL**

HSSCCH\_PART2：有一个CRC，**只打印一个LABEL**

AGCH：有两个UE\_ID，对应两个CRC结果，如果第一个校验错误，HW会尝试第二个UE\_ID，所以如果第一个UE\_ID校验通过，只打印一个LABEL，否则第二个UE\_ID对应的unmasked CRC值和CRC校验结果也需要打印

打印Label（两个）：

XXXX\_CRC\_PAR

DataLength 2 BitWidth 16

XXXX\_CRC\_SEC

DataLength 2 BitWidth 16

打印格式：

第一个十六进制数表示unmasked CRC，第二个十六进制数表示CRC校验结果

打印示例

0x1234 0x0001

**圆序号9**

节点说明：

CRC校验通过时，需要再计算SER，进行false alarm判定。SER计算过程是，解码比特再编码，和接收软比特符号位进行对比，使用**错误软比特的数量作为SER值**。

打印要求

HSSCCH\_PART1：始终计算SER，打印LABEL

HSSCCH\_PART2：CRC正确时才计算SER，打印LABEL，否则不打印

AGCH：CRC正确时才计算SER，打印LABEL，否则不打印

打印Label：

XXXX\_SER

DataLength 1 BitWidth 7

打印格式：

十六进制数SER

打印示例

0x12

## 1.3 测试平台对比需求

相对于V2 VDEC\_HSSCCH平台，V3增加以下vdec\_hs内部对比节点

1. 软比特unmask节点

LOG LABEL：

XXXX\_UNMASK

抽取说明：

此LABEL对应的数据全部抽取到单独文件，每次对比一个软比特，所以不用合并

对比条件：

下述信号位于vdec\_hs\_inst.u1\_fwd

if (crc\_en == 1 && cyc[6:0] == 7’d2) begin

// 对比三个软比特，信号c0对应第一个接收到的软比特，c1，c2对应第二和第三个。

end

1. State metric节点

LOG LABEL：

XXXX\_STATE\_METRIC

抽取说明：

此LABEL对应的数据全部抽取到单独文件，每次对比四个STATE\_METRIC

对比条件：

下述信号位于vdec\_hs\_inst.u1\_fwd

if (sm0\_wr == 1) begin

// 取四个state metric和信号sm0\_din对比

end

else if (sm1\_wr == 1) begin

// 取四个state metric和信号sm1\_din对比

end

1. Survive path节点

LOG LABEL：

XXXX\_SURVIVE\_PATH

抽取说明：

此LABEL对应的数据全部抽取到单独文件，每次对比32比特

对比条件：

下述信号位于vdec\_hs\_inst.u1\_fwd

if (pt\_wr == 1) begin

// 取一个32bit word和信号pt\_din进行对比

end

1. Traceback state节点

LOG LABEL：

XXXX\_TRACE\_STATE

抽取说明：

此LABEL对应的数据全部抽取到单独文件，每次对比一个8比特word

对比条件：

下述信号位于vdec\_hs\_inst.u1\_bwd

if (pt\_rd == 1) begin

// 取一个8 bit word和信号pre\_state进行对比

end

1. Decode bits节点

LOG LABEL：

XXXX\_DEC\_BITS

抽取说明：

此LABEL对应的数据全部抽取到单独文件，每次对比一个32比特word

对比条件：

下述信号位于vdec\_hs\_inst.u1\_bwd

if (done == 1) begin

// 取一个32 bit word的低29bits和信号dec\_bits进行对比

end

1. CRC节点

LOG LABEL：

XXXX\_CRC\_PAR

XXXX\_CRC\_SEC

抽取说明：

这两个LABEL对应的数据全部抽取到单独文件，每次对比一行，两个16比特word

对比条件：

下述信号位于vdec\_hs\_inst

if (crc\_done == 1) begin

// 先取一个16bit word和crc\_par\_bits对比

// 再取一个16bits word的最低位和crc\_match对比

end

1. SER节点

LOG LABEL：

XXXX\_SER

抽取说明：

此LABEL对应的数据全部抽取到单独文件，每次对比一个7比特word

对比条件：

下述信号位于vdec\_hs\_inst

if (ser\_done == 1) begin

// 取一个8 bit word的低7bit和信号ser\_acc进行对比

end