

### 1.1.1.1 Tail biting viterbi 的译码

LTE的viterbi编码是tail biting的，没有尾比特，所以译码的时候要对branch metric数据做前后扩展(如下图)。译码后的结果要截取中间部分的hardbits。下面是一个例子：

假设编码的比特数是56，编码速率是1/3，编码器约束是7(6级延迟寄存器)

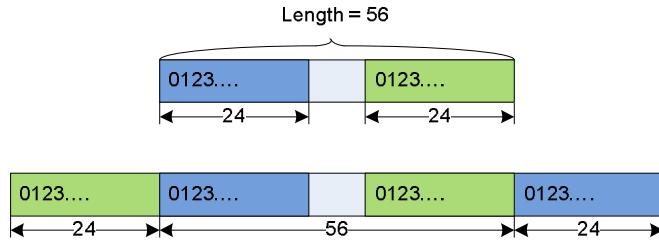
编码比特流是 0x12, 0x34, 0x56, 0x78, 0x9a, 0xbc, 0xde,

计算出来的 branch metric 的 size 是  $56 \times 4$ .

前扩展的 size 是:  $4(\text{BM values per stage}) * 6(\text{delay register}) * 4$  (4倍约束)

前扩展的 size 是:  $4(\text{BM values per stage}) * 6(\text{delay register}) * 5$  (5倍约束)

如下图:



编码器约束是7，编码速率是1/3，所以每个stage的branchmetric有4个softbits  
所以patch的时候首尾各加上 $(7-1)*4=24$

译码的输出如下图:

```
0x00813EE0  vcpHdBuf
0x00813EE0  9A BC DE 12 34 56 78 9A BC DE 12 34 57
```

除掉前扩展的 24 个比特，结果就是 0x12, 0x34, 0x56, 0x78, 0x9a, 0xbc, 0xde