

## Comparison of AN decoder

為了要觀察是否使用 **barrett reduction** 是否能夠有效降低硬體成本，做了這份報告。

### 前言

要先了解 **coding style** 對 **synthesis** 的影響有多大觀察以下兩個 **code**，這兩種 **code** 計算結果相同，但硬體合成時，**timing LUT** 數量都顯著的不同

Case1

```
```C
```

```
q=x*12345>>45
```

```
```
```

Case2

```
```C
```

```
q=x*12345
```

```
q=q>>45
```

```
```
```

|       | DSP | FF  | LUT |
|-------|-----|-----|-----|
| Case1 | 5   | 331 | 244 |
| Case2 | 0   | 0   | 89  |

相同計算，但僅僅只是換了行就能夠不使用 **DSP**。

## Barrett reduction/乘法餘數比較

下表為 **A=29** 時，使用

1. **barrett reduction** 當作 **decoder** 的硬體資源是 **case1**
2. **barrett reduction** 當作 **decoder** 的硬體資源是 **case2**
3. 使用乘法(\*)和餘數(%)

|   | DSP | FF  | LUT |
|---|-----|-----|-----|
| 1 | 5   | 331 | 244 |
| 2 | 0   | 0   | 89  |
| 3 | 3   | 188 | 219 |

## TCB 與乘法的比較

| Test code   | Interval | DSP | FF  | LUT |
|---|----------|-----|-----|-----|
| <pre>void mul_5(int &amp;a){     a=a*5; }</pre>   | 1        | 0   | 0   | 39  |
| <pre>void TCB_mul_5(int &amp;a){     a=(a&lt;&lt;2)+(a); }</pre>  | 1        | 0   | 0   | 39  |
| <pre>void normal_mul(int *arr){     for( int i=0;i&lt;10;i++){         *(arr+i)=*(arr+i)*weight[i];     } }</pre>   | 24       | 1   | 195 | 146 |
| <pre>void TCB_mul(int *arr){     for( int i=0;i&lt;10;i++){         *(arr+i)=TCBop(*(arr+i),weight[i]);     } }</pre>   | 13       | 0   | 12  | 322 |
| <pre>int TCBop(int val,int weight){     switch(weight){     case 1:         return(val);     case 2:         return(val&lt;&lt;1);     case 3:         return((val&lt;&lt;1)+(val));     case 4:</pre>  |          |     |     |     |
| <pre>    //int weight[]={1, 1, 7, 1, 4, 2, 4, 1, 9, 2};     void TCB_table(int *arr){         int temp=*(arr);         *(arr)=temp;         temp=*(arr+1);         *(arr+1)=*(arr+1);         temp=*(arr+2);         *(arr+2)=(temp&lt;&lt;3)-temp;</pre> | 無法合成     |     |     |     |

## 討論

目前還在思考如何將 TCB 寫成 hls style 的 code，不太明白如何以高階語言寫成 TCB。

或許可以更改.v 中的 code，但 verilog code 都被包成一層一層的硬體 IP，難以修改。