

# NPM 解読 報告

2015.10.02 Shunichi Akatsuka

# 全体の状況

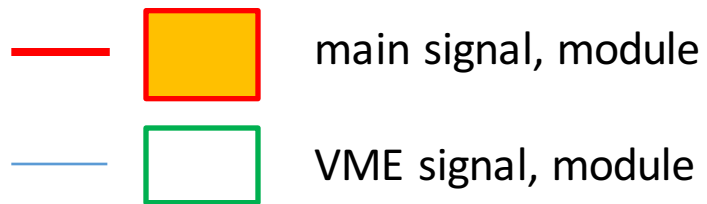
9/18 Fri. NIM Process Module (NPM)の解読を始めた。

9/29 Tue. 進捗報告をして、その際に疑問点が解決。

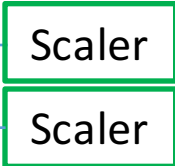
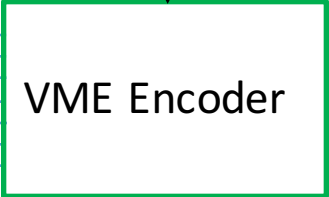
9/30, 10/1 Vivado でコードを見て最終的な理解をまとめた。

10/2 Fri. ←本日。発表します。

# NPMの流れ



NPMは入力信号に様々な操作をし、加工して出力するmodule



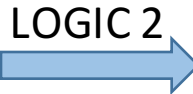
INPUT [75:0]



FPGA\_OUT0

FPGA\_OUT1

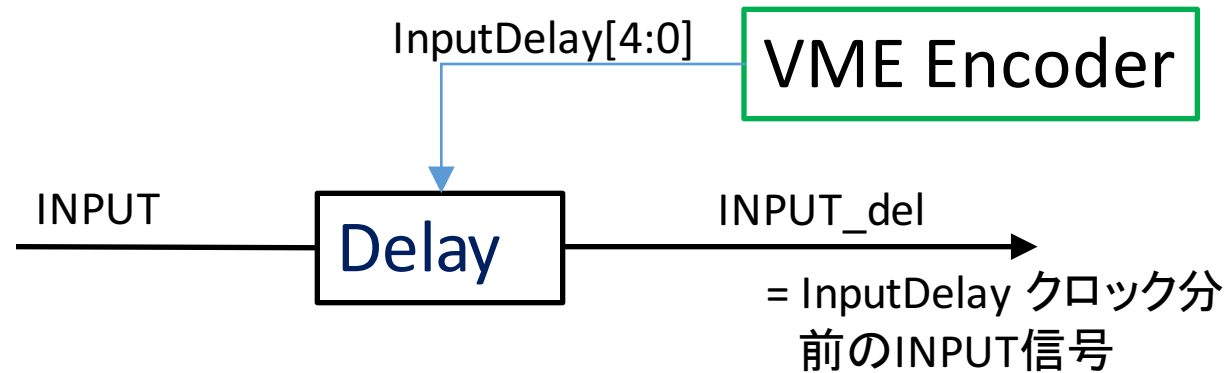
2つの経路に分ける！！



# Delay, Mask

- Delay module

→ VME から delay 値を受け取り、入力信号に適用し出力するモジュール



- Mask

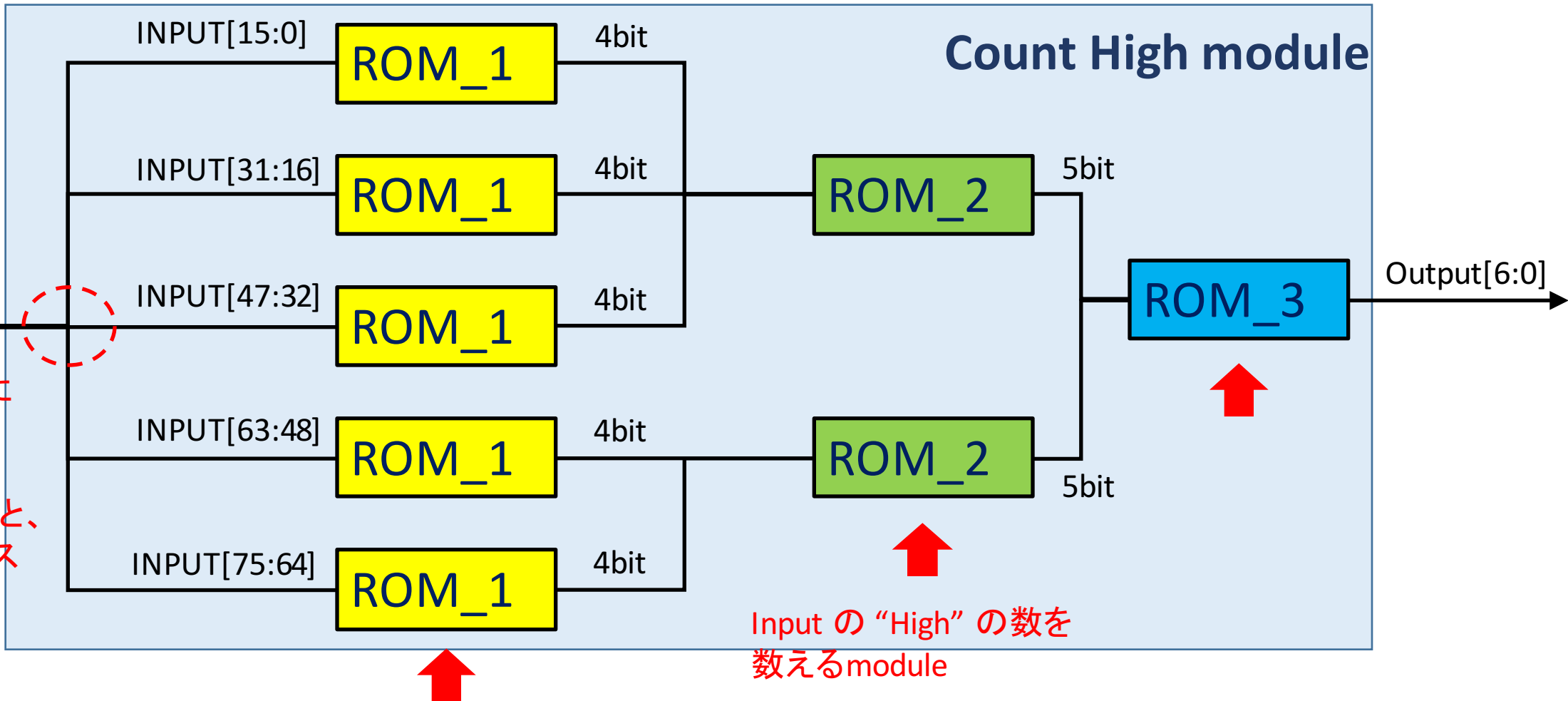
→ 見たくない信号を Mask してしまう

```
assign INPUT[0] = InputMask1[0] ? 1'b0 : INPUT0_del;
```

↑  
InputMask が “1” → INPUT に 0 を代入  
InputMask が “0” → そのまま

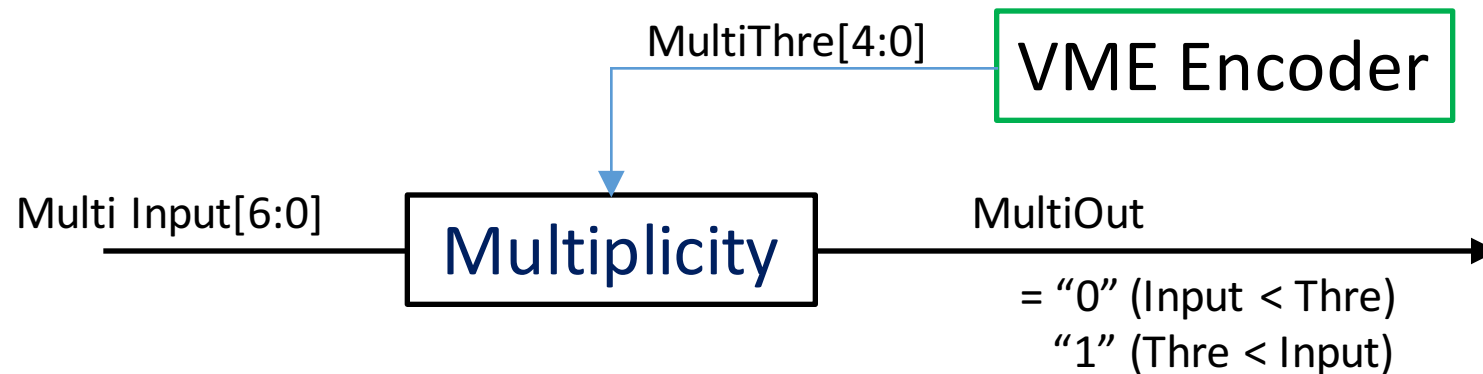
# Count High

- INPUT [75:0] の76bitのうち**幾つが”1”**かを数え、Output[6:0]を返す。



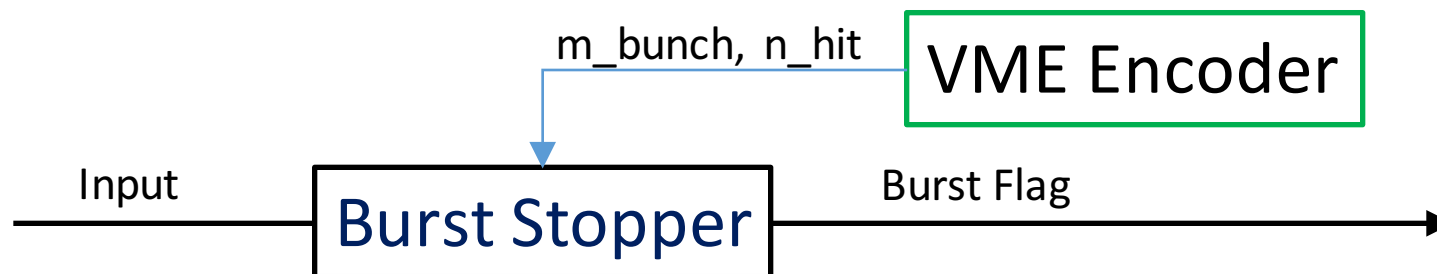
# Multiplicity

- Count High の output が threshold を越えていれば “1”, 越えていなければ “0” を出す。
- Threshold はVME を用いて受け渡すことができる。



# Burst Stopper

- 直近のinput  $m$ 個のうち、 $n$ 回以上がHigh だとBurst Flag を出す。



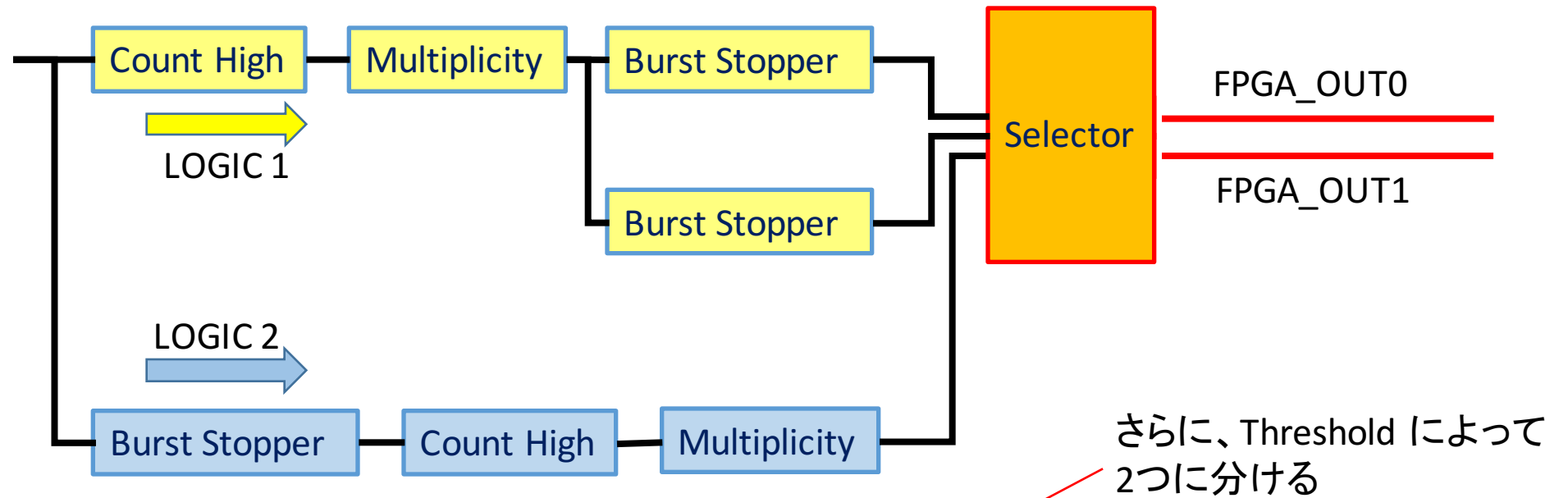
**Logic** [ $m=6, n=3$  の場合]

x	010011010010	1 <-input	hit_num = 2	BurstFlag = "0"
	出たく	入る	↓ +1	
0	100110100101	0 <-input	hit_num = 3	BurstFlag = "1"
			↓ -1	
1	001101001010	0 <-input	hit_num = 2	BurstFlag = "0"

緑の数字が大事で、"m bunch" の中に  
入ってくるのが "1", 出て行くのが "0" なら hit\_num は 1増加  
入ってくるのが "0", 出て行くのが "1" なら hit\_num は 1減少

# Selector

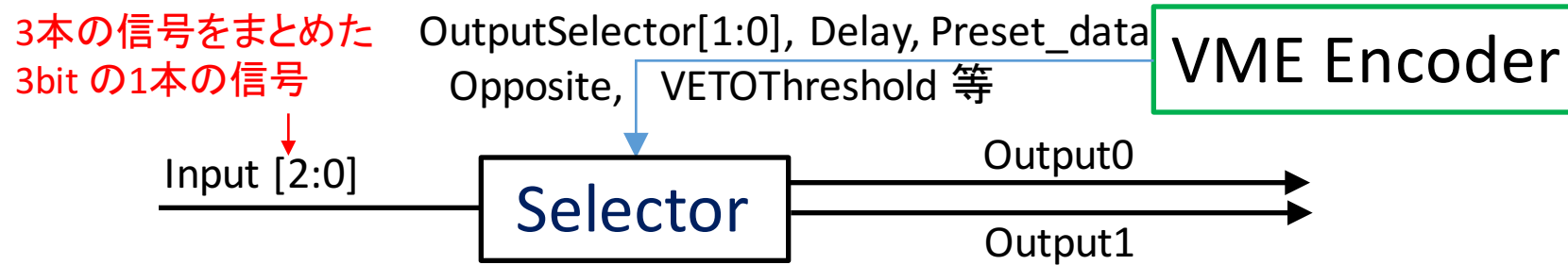
- これまで説明した module を使って作った2つの経路からのどの input を使うかを選択する



- LOGIC 1: Input のうち、High のものが多い状態が続くと BurstFlag
- LOGIC 2: High の状態が続いているInput が多いと BurstFlag



# Selector



- UseOpposite = “1” → 逆サイドの SL を使う  
= “0” → OutputSelector の値で使う Input のbit数 指定
- Delay に対応した分 信号を遅らせる (Opposite とタイミング合わせる)

# Selector

- Trigger (Output0)
  - Burst 中のイベント情報のsampleが少し欲しいので、Burst 中たまに trigger を発行するようになっている
  - 具体的には、Burst の回数をCounterに記録し、VETOThresholdを越えたら trigger を一回だけ出し、Counterをresetする。
- VETO (Output1)
  - Burst が起こったらVETO をかける。
  - 一度 VETO を出したら、Preset\_data 回は出し続ける。

# Scaler, Burst Scaler

- Scaler

- INPUT が“1” になった回数を数える module
- Output はVME へ

- Burst Scaler

- BurstFlag が “1” になった回数を数える -> VETO bunch
- 同時に、“BurstFlagの塊” が何回あったかも数える -> Burst Count

- 例:

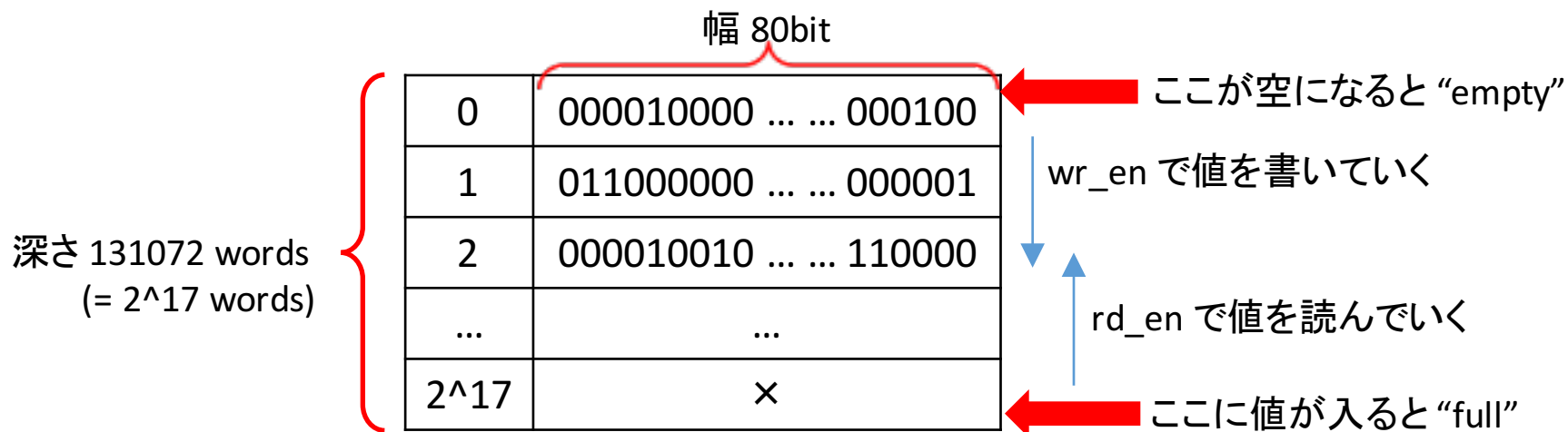
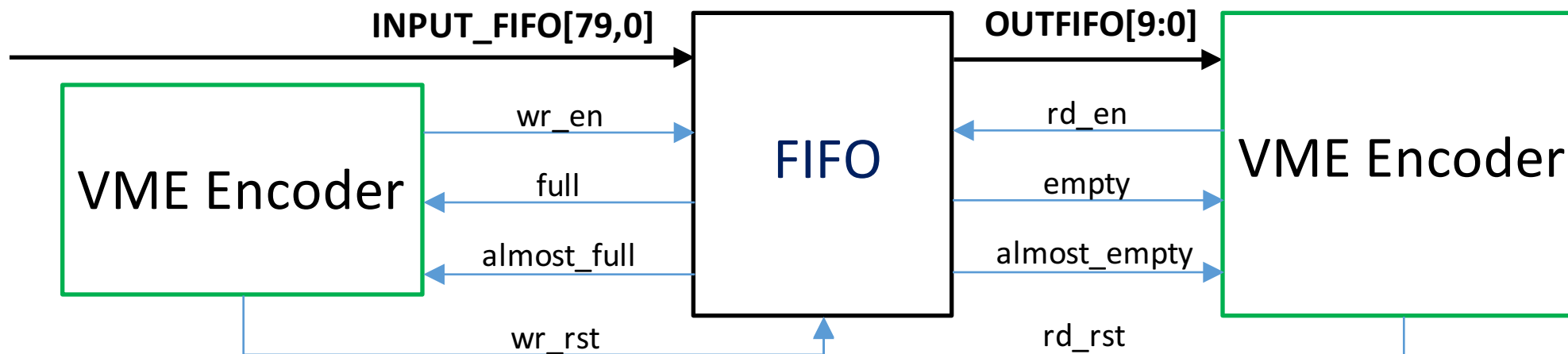
BurstFlag: 00000111100011100110000000

VETO bunch = (“1” の数) = 9

Burst Count = (“塊” の数) = 3

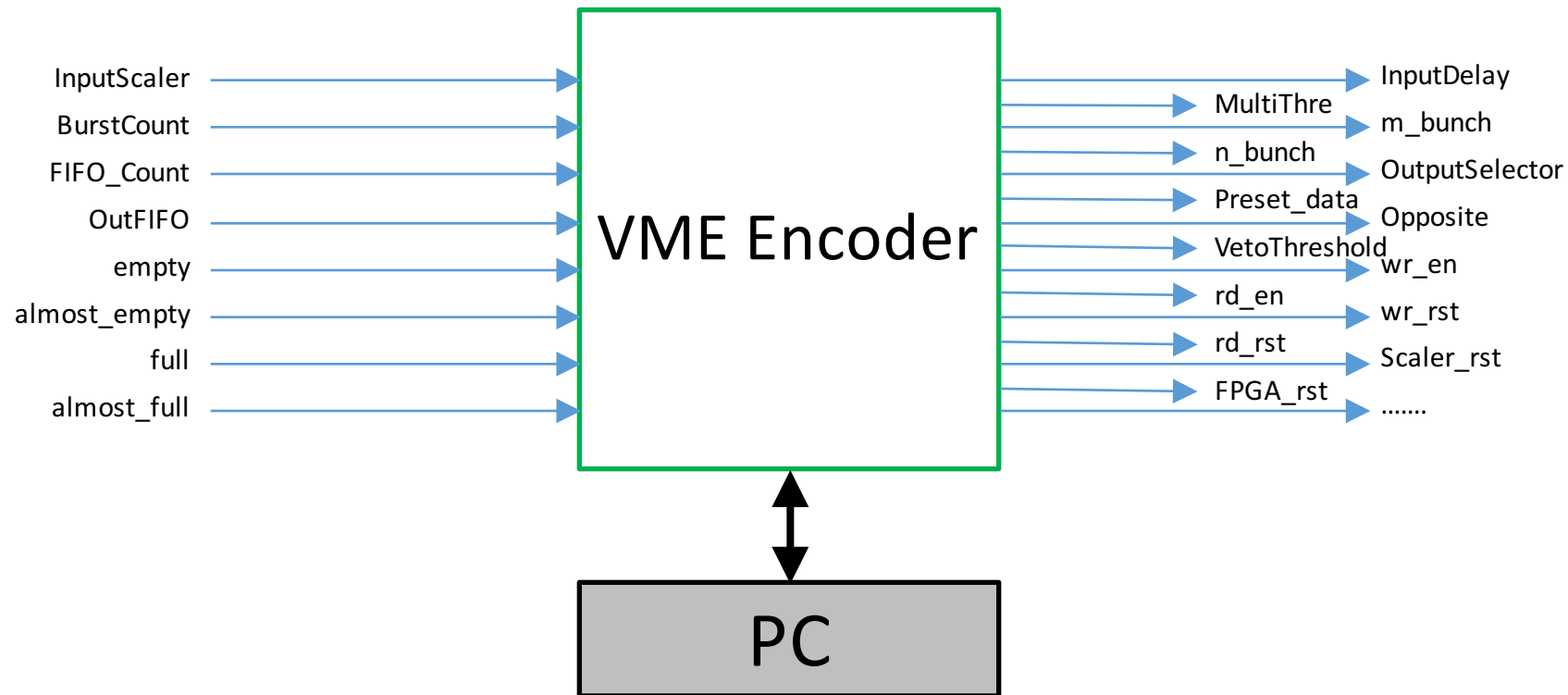
# FIFO

- First In, First Out --> input を保持し、読み出す。



# VME Encoder

- Delay値, Threshold, n/m の値等、様々な値を各module に与え、また Scaler の値やFIFO の Output 等を受け取る。
- これらの値はPC を使って制御・変更できる。



# まとめ、他

- NPMの firmware はほぼ解読できた。
- Vivado で見ないとみえない情報(FIFO, ROM等)もある。
- 来週からは、現行の SL を読む？
  - 10月末にテスト
- その他
  - 自分のdynabook -> Vivado 2015 が入っていて、今これで見ている
  - 先日もらった Let's note -> Vivado 2014. Update したいけど、アカウント情報をなんとか、と言われる。
  - 東棟160 のwindows -> Vivado が動かない、、