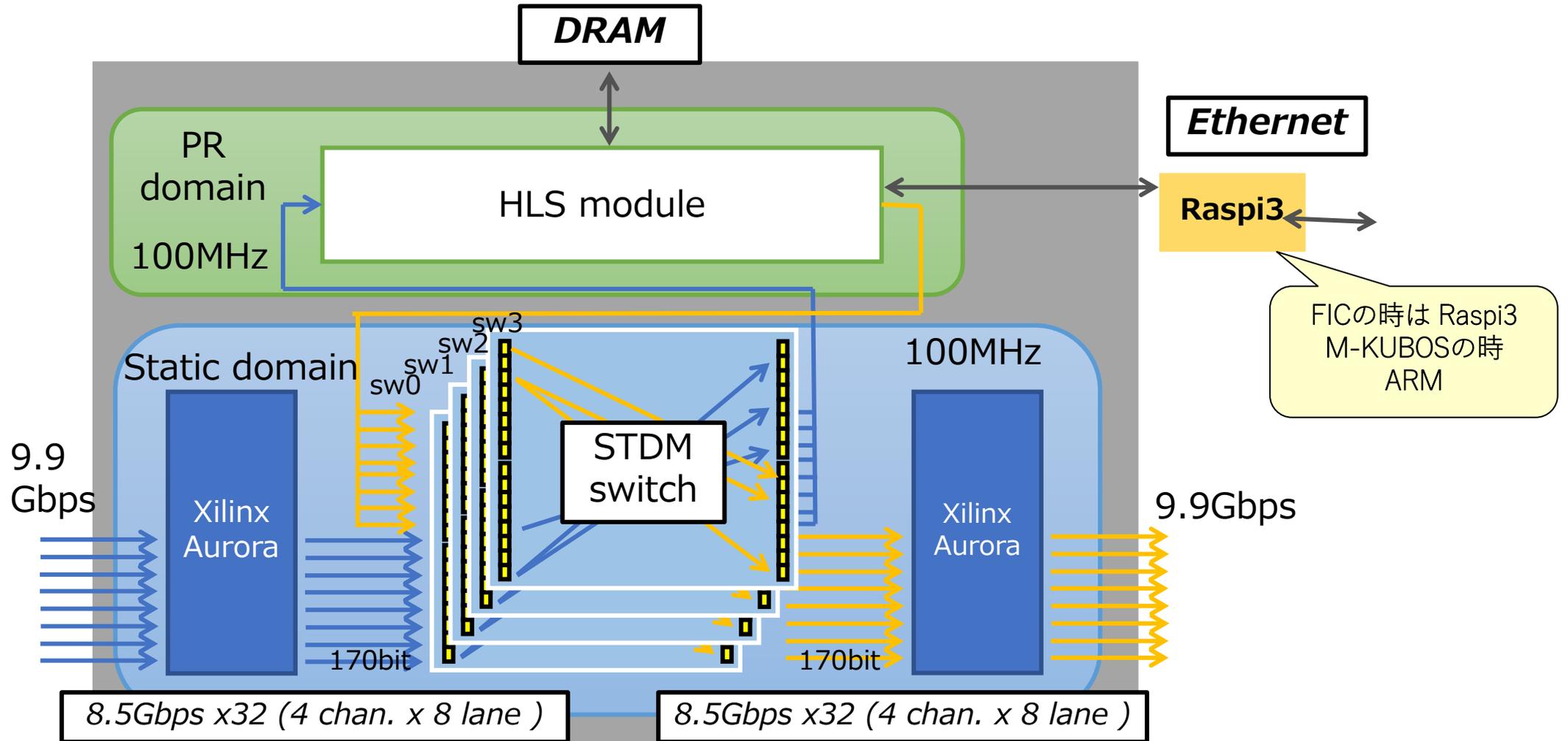


Crestサイトビジット

若林グループ（東大＋NEC）

大規模なプログラムを自動分割して
複数のFPGAで実行する設計ツールの研究開発

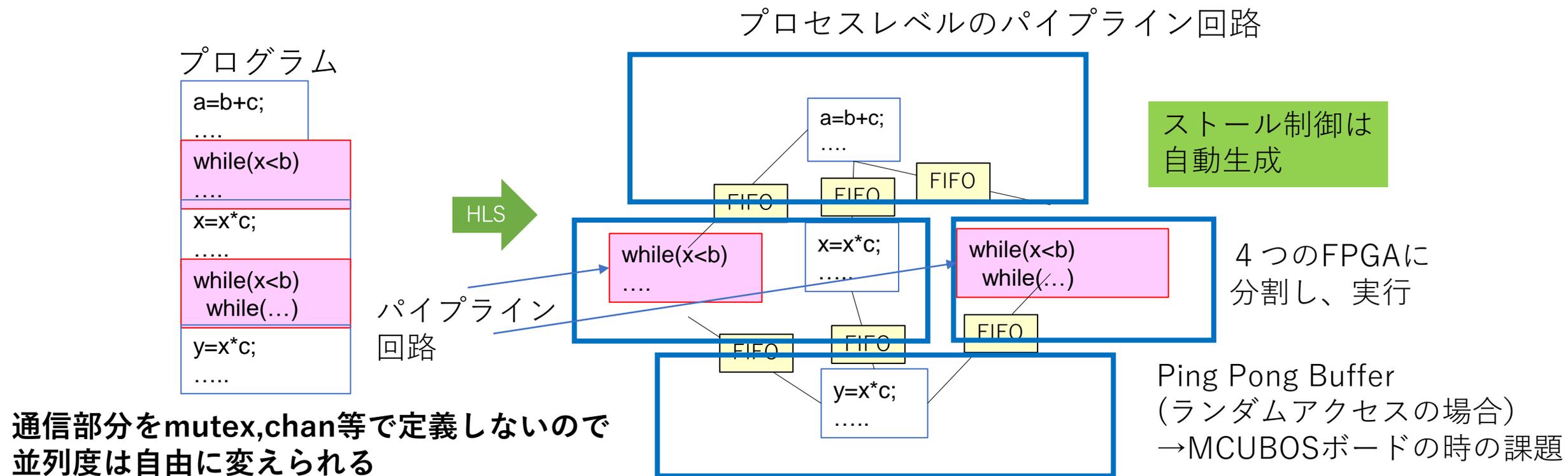
FIC (またはM-KUBOS) ボードの構成



C/C++プログラムを分割し、各ボードにマップし
データをパケット化して、送受信できる設計環境を作る

逐次記述C++を、データフロー計算（自動並列パイプライン化）

FPGAと高位合成ツールを利用して、汎用的なデータフロー計算機を構築する
複数FPGAを利用することで、大規模プログラムも深いパイプラインで実行可能



回路の自動分割と、FPGAボード間の通信回路の自動生成

- データをシーケンシャルアクセスする場合は分割だけでOK.
ランダムアクセスは、共有メモリが必要で、改善が必要

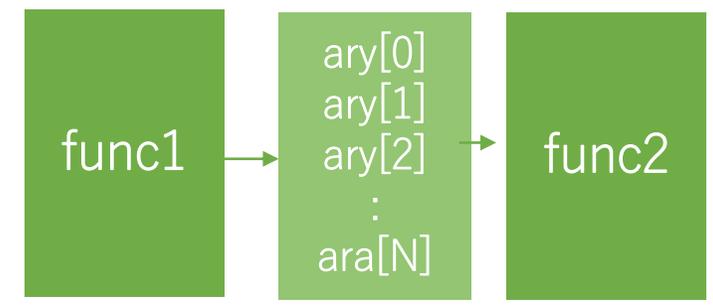
分割ポイントの例

プロセスレベルの並列性=> **順次アクセス**とわかれば、自動FIFO化

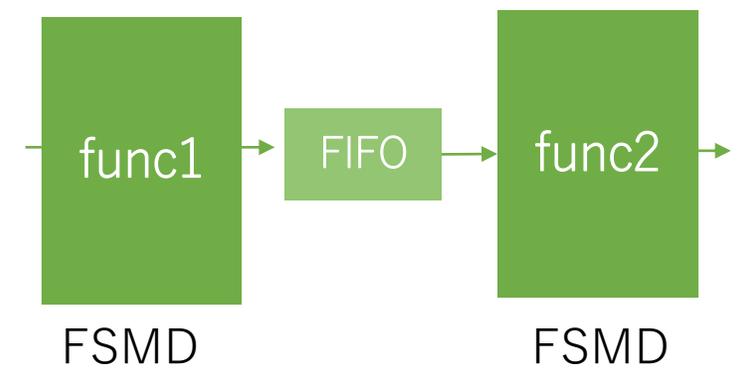
```
func1(){  
  //cyber foldings=1  
  for (int i = 0; i < N; i++) {  
    ary[i] = a0[i]+.....;  
  }  
}  
  
func2(){  
  //cyber foldings=1  
  for (int i = 0; i < N; i++) {  
    x0[i] = ary[i];  
  }  
}  
//cyber process_pipeline  
Main(){  
  func1();  
  func2();  
}
```

ボード分割に適した部分

単純合成だと、SRAM必要だが、



SRAM消去
FIFOに自動変換

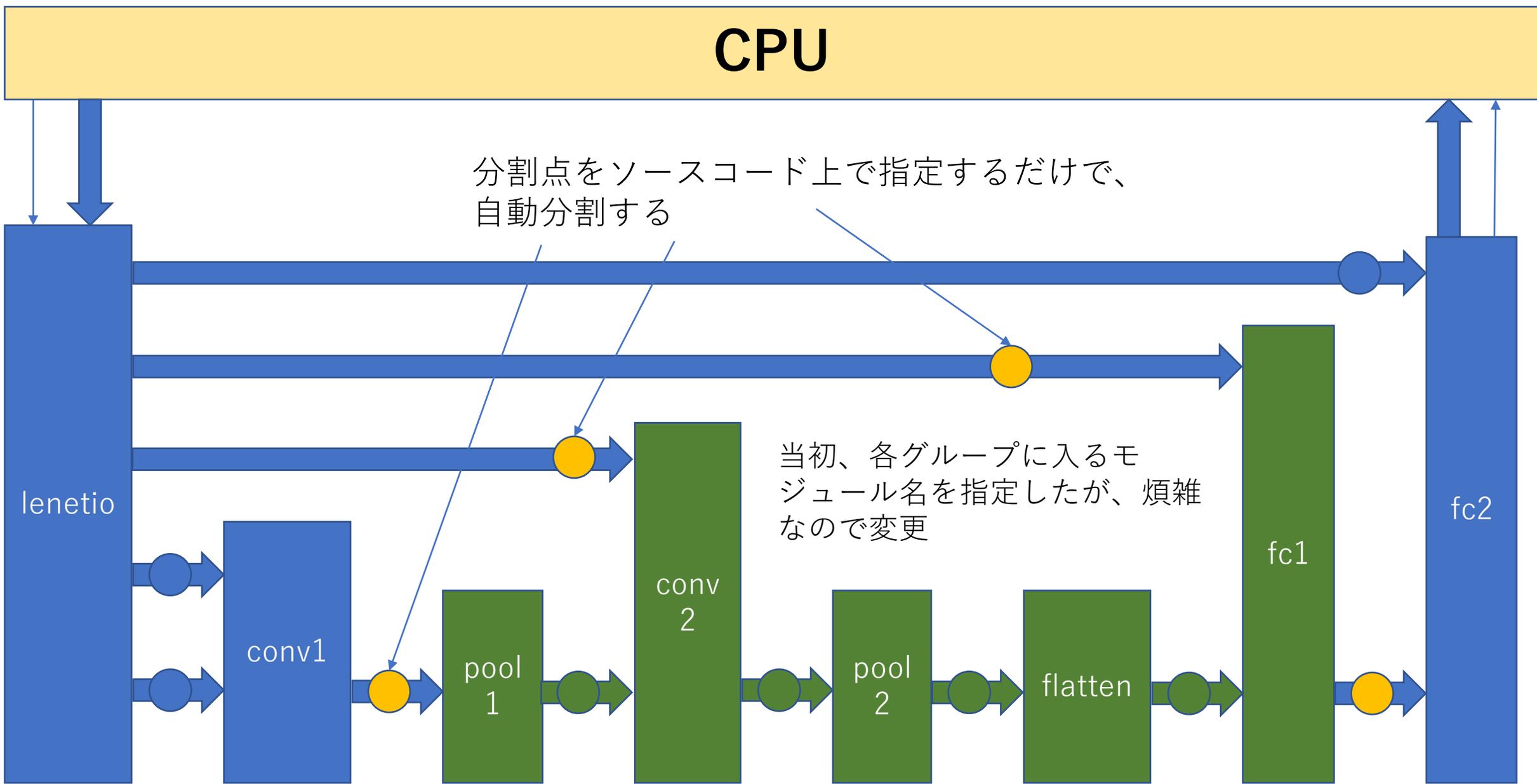


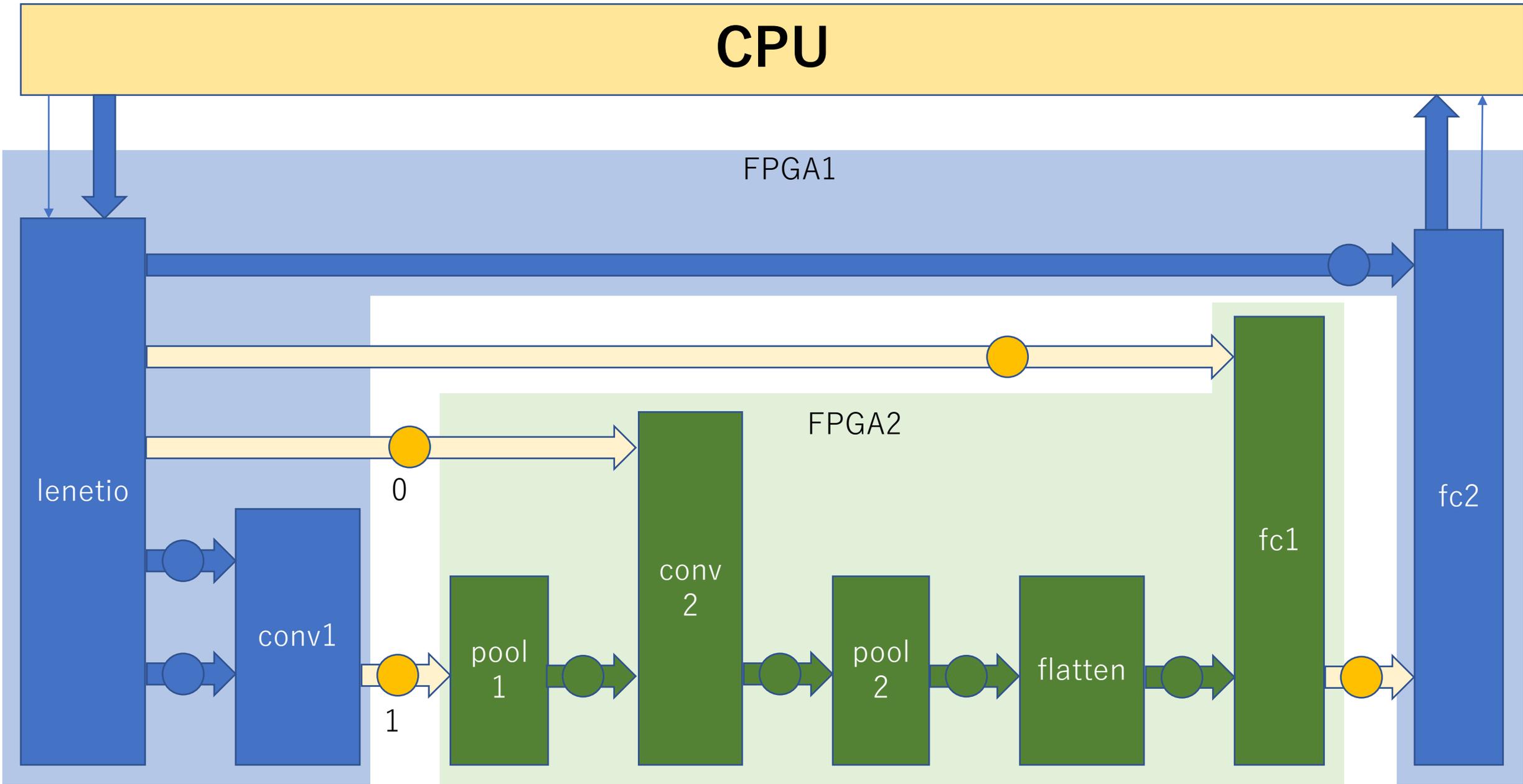
ランダムアクセスならば、PingPongバッファに自動変換可能

CPU

分割点をソースコード上で指定するだけで、自動分割する

当初、各グループに入るモジュール名を指定したが、煩雑なので変更





S0から来たデータの上位ビットには0を付加して
S1から来たデータの上位ビットには1を付加して
送信する
同サイクルに来た時は優先順位を決めて
どちらかだけを送る

CPU

FPGA1

受信したデータの上位ビットが0の場合はr0に送信
受信したデータの上位ビットが1の場合はr1に送信

lenet
io

co
nv
1

s0

s1

送信
回路

受信
回路

r0

r1

co
nv
2

FPGA2

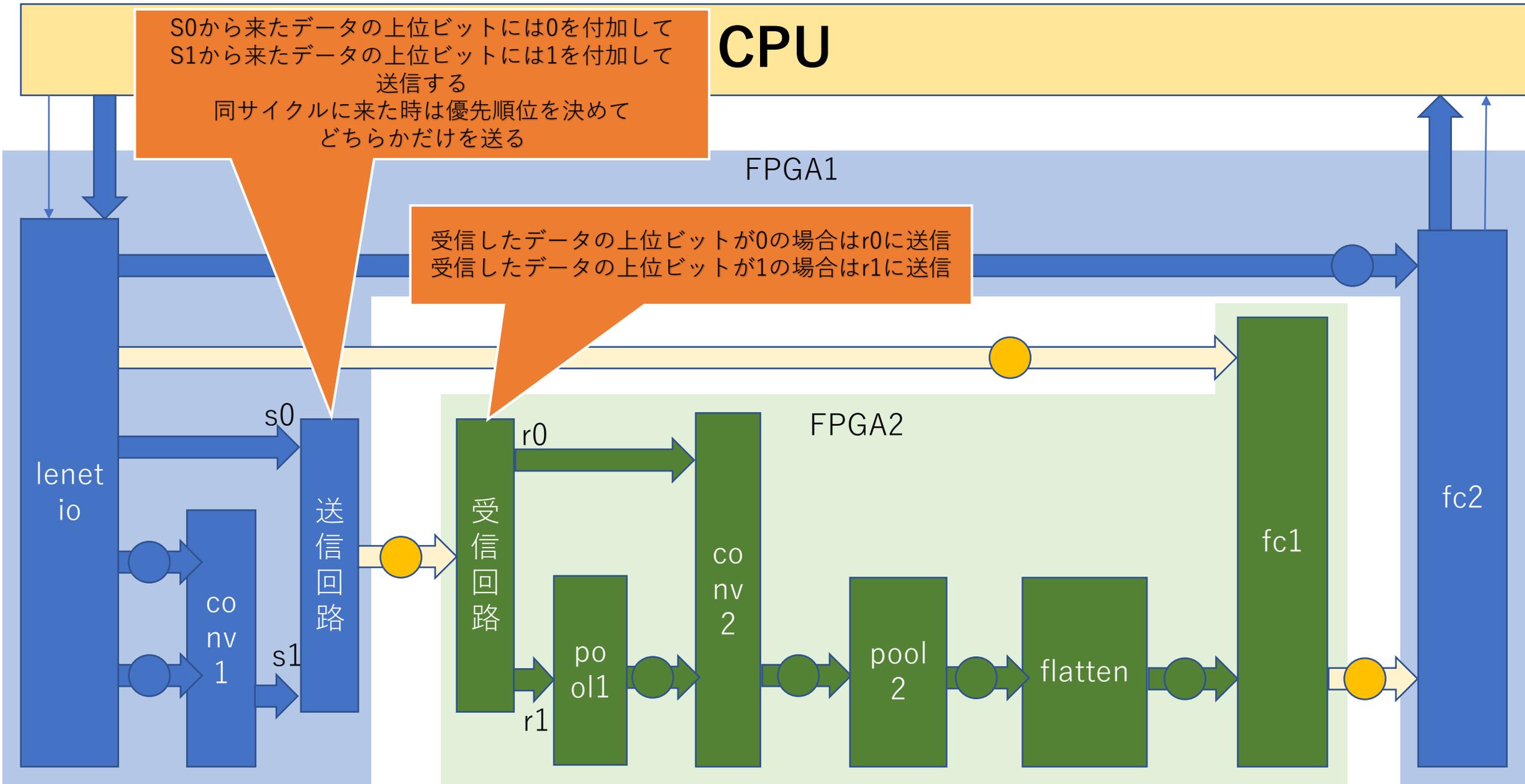
po
ol
1

po
ol
2

flatten

fc1

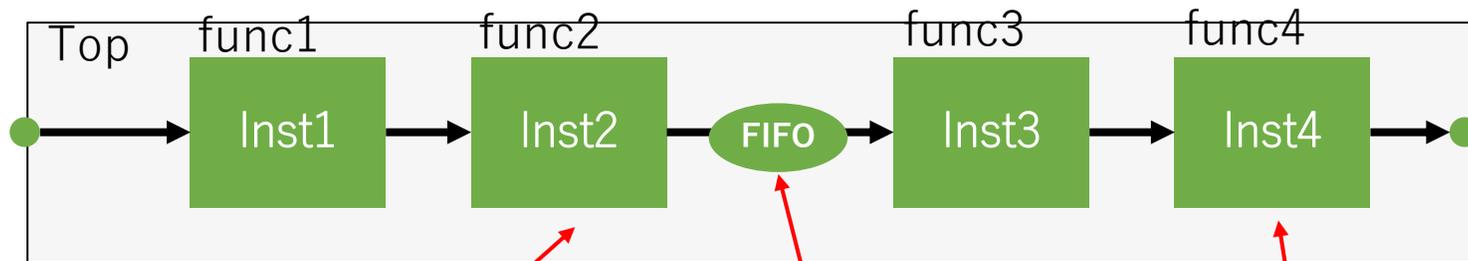
fc2



手動分割

- 分割は、回路規模、通信容量、FPGAボードが実際に結合されているリンクの関係（例：リング構造、メッシュ構造）等を考慮して決める必要がある。
自動化は難しいため、現在は人手で、分割点を指定
- 人手でも分割点の最適化決定は難しいので、試行錯誤が行いやすいように設計

モジュール分割機能向けGUI



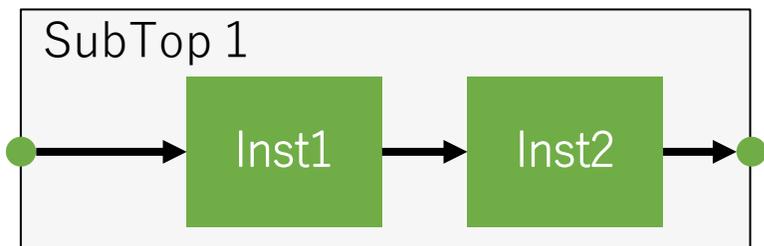
module_group = SubTop1

partition=4

module_group = SubTop2

アトリビュートで各インスタンスの分割先を指定する(将来的には別ファイル指定も可能とする)

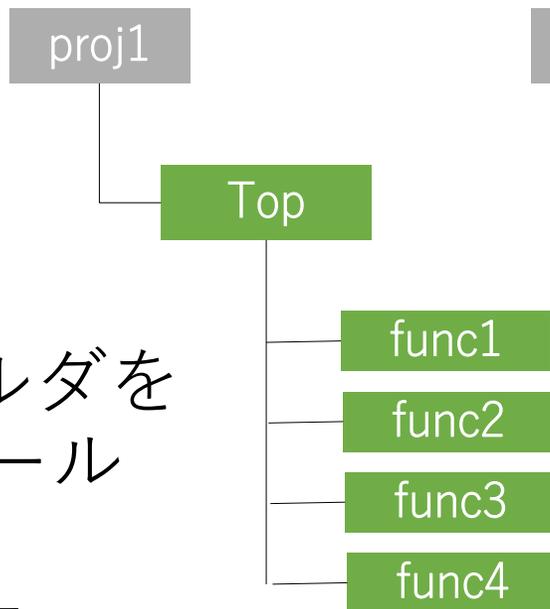
```
bdltran -Zmodule_divide=SubTop1 Top.IFF ...
```



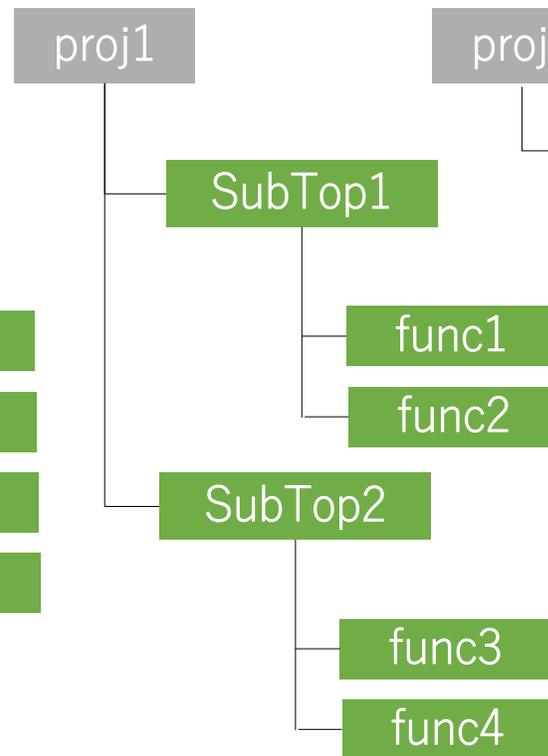
```
bdltran -Zmodule_divide=SubTop2 Top.IFF ...
```



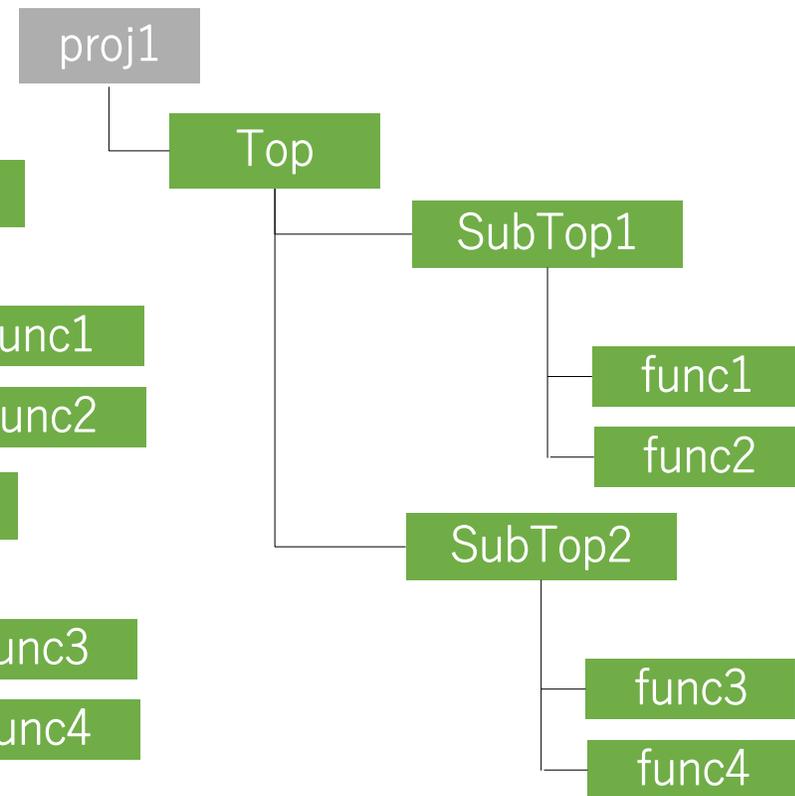
A.分割なし



B.分割あり

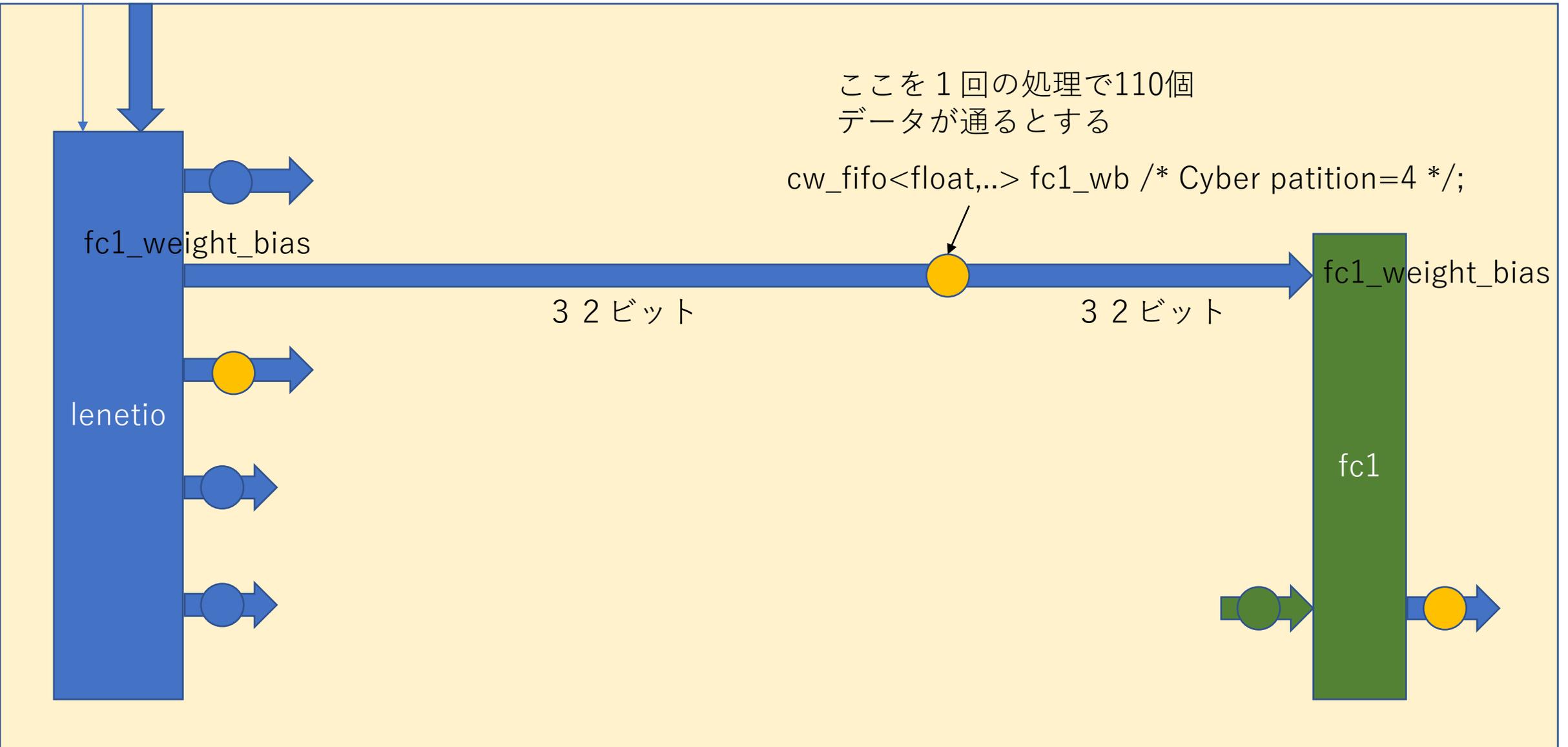


C.分割あり (将来必要になったら)



- parser 直後はAで、Topのプロセスフォルダを右クリックでモジュール分割を実行すると
 1. モジュール分割判定
 2. 各モジュールSubTop#を全部合成が実行されてBになる。

分割前



分割後

ここを1回の処理で110個
データが通るとする

32ビット

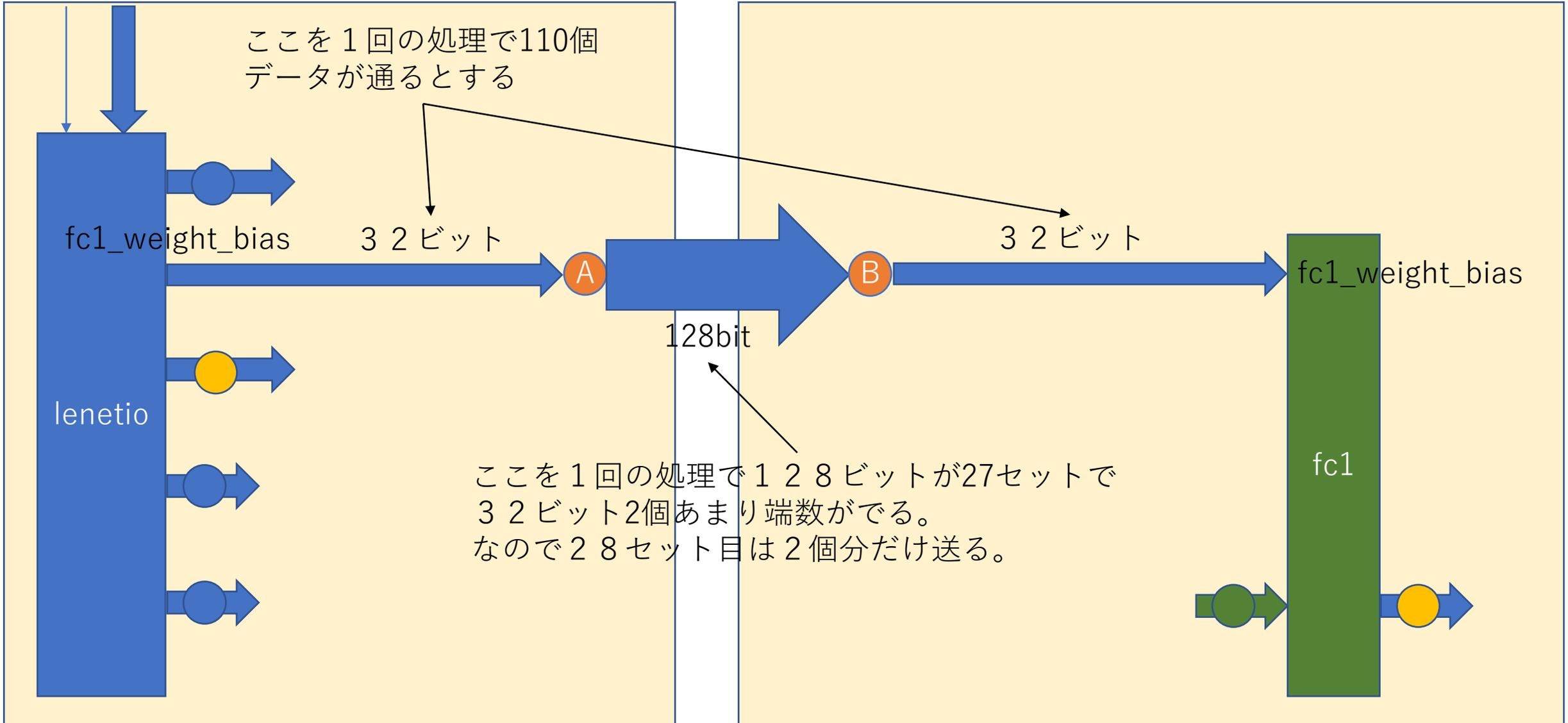
A

128bit

B

32ビット

ここを1回の処理で128ビットが27セットで
32ビット2個あまり端数がでる。
なので28セット目は2個分だけ送る。



案 1

```
cw_fifo<float,..> fc1_wb /* Cyber patition=4,data_flow=110 */;
```

ここを1回の処理で110個
データが通るとする

1回の処理で何個データが通るか
ユーザが指定する

32ビット

32ビット

128bit

fc1_weight_bias

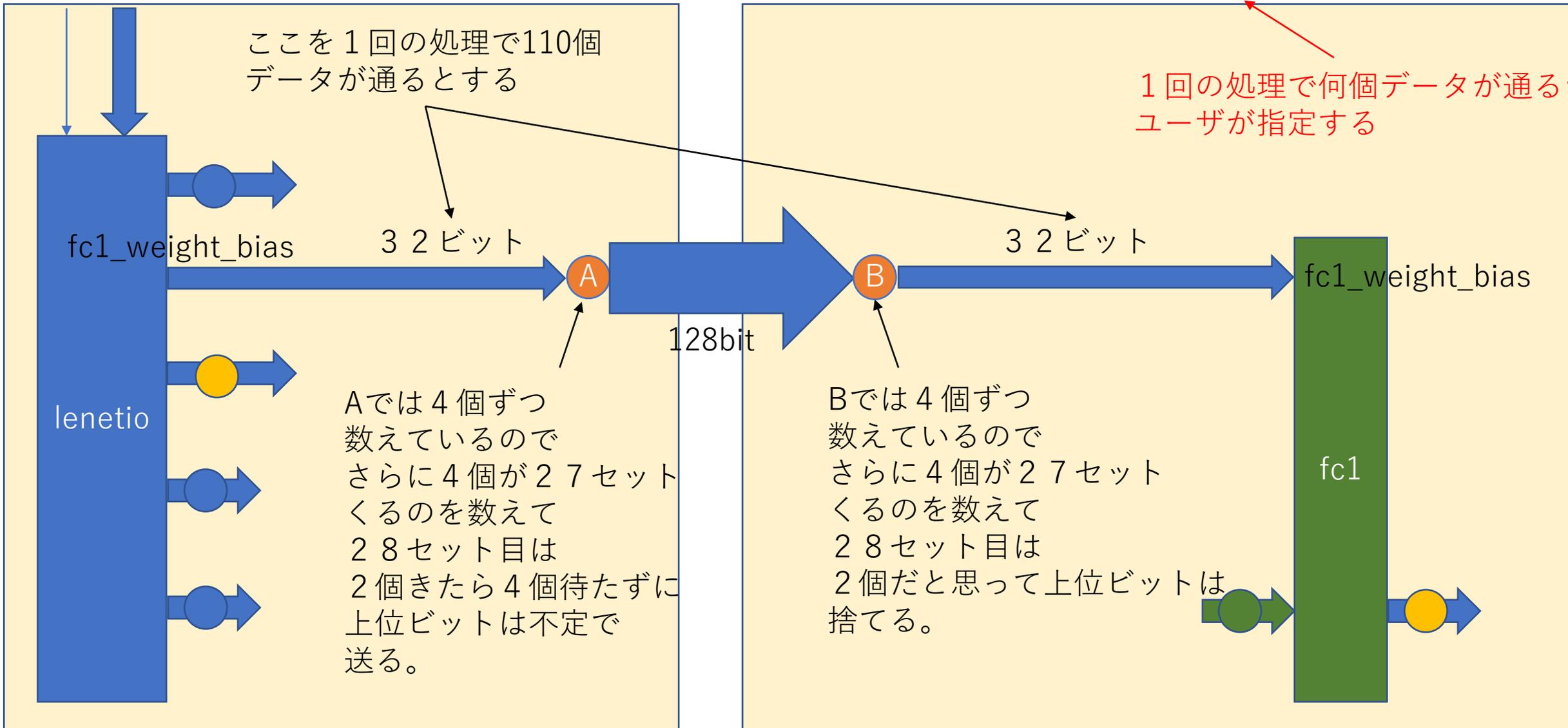
fc1_weight_bias

lenetio

fc1

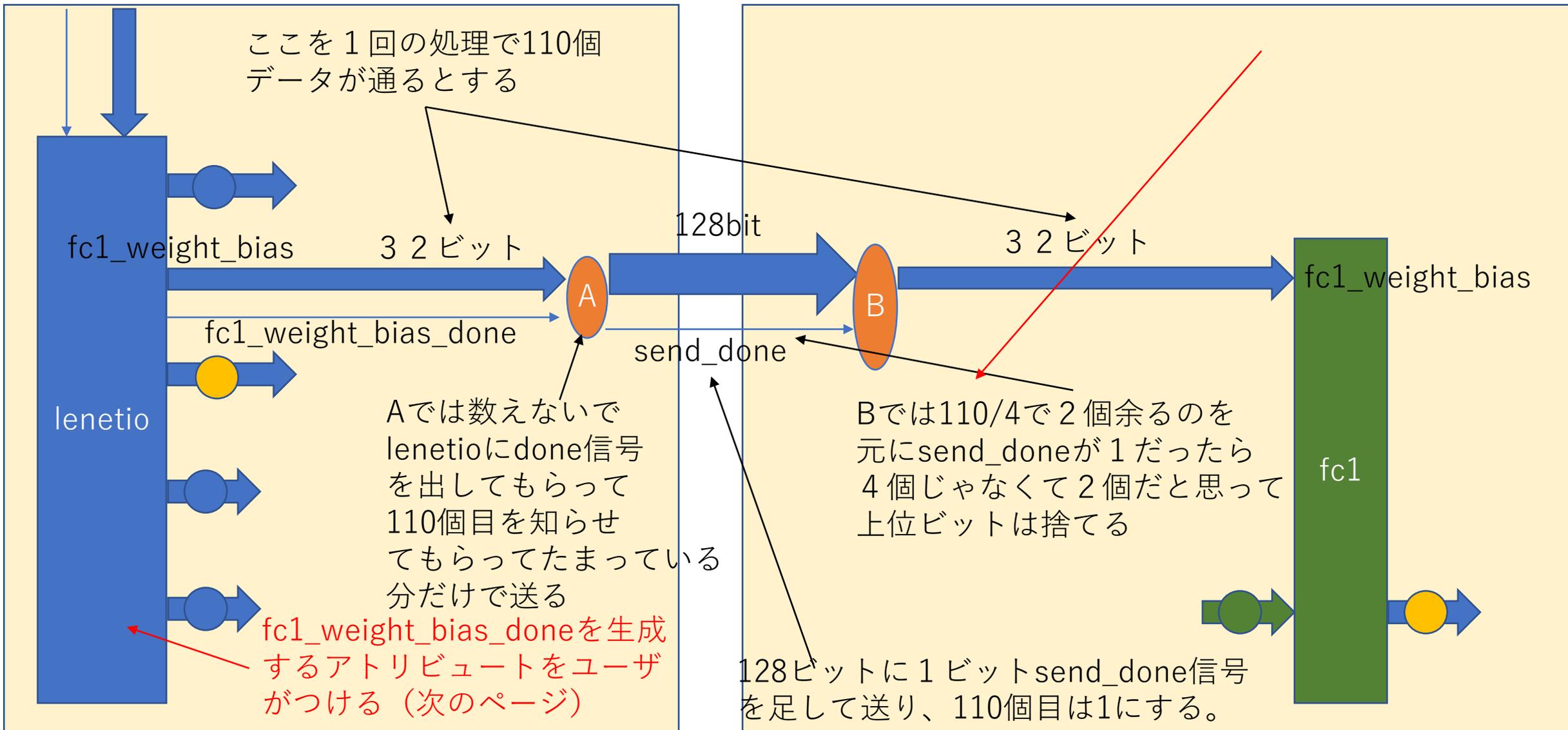
Aでは4個ずつ
数えているので
さらに4個が27セット
くるのを数えて
28セット目は
2個きたら4個待たずに
上位ビットは不定で
送る。

Bでは4個ずつ
数えているので
さらに4個が27セット
くるのを数えて
28セット目は
2個だと思って上位ビットは
捨てる。



案 2

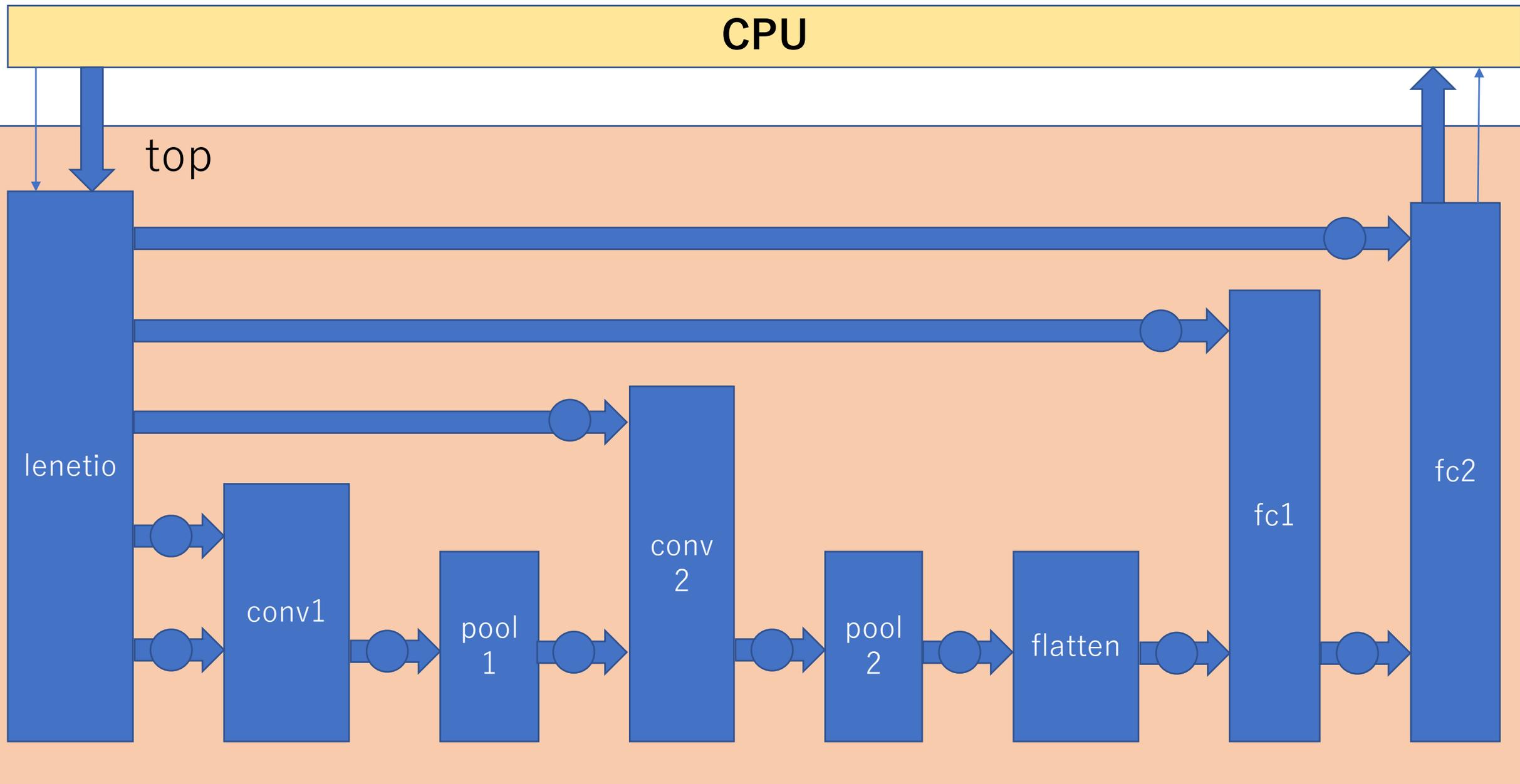
```
cw_fifo<float,..> fc1_wb /* Cyber patition=4,data_flow=110*/;
```



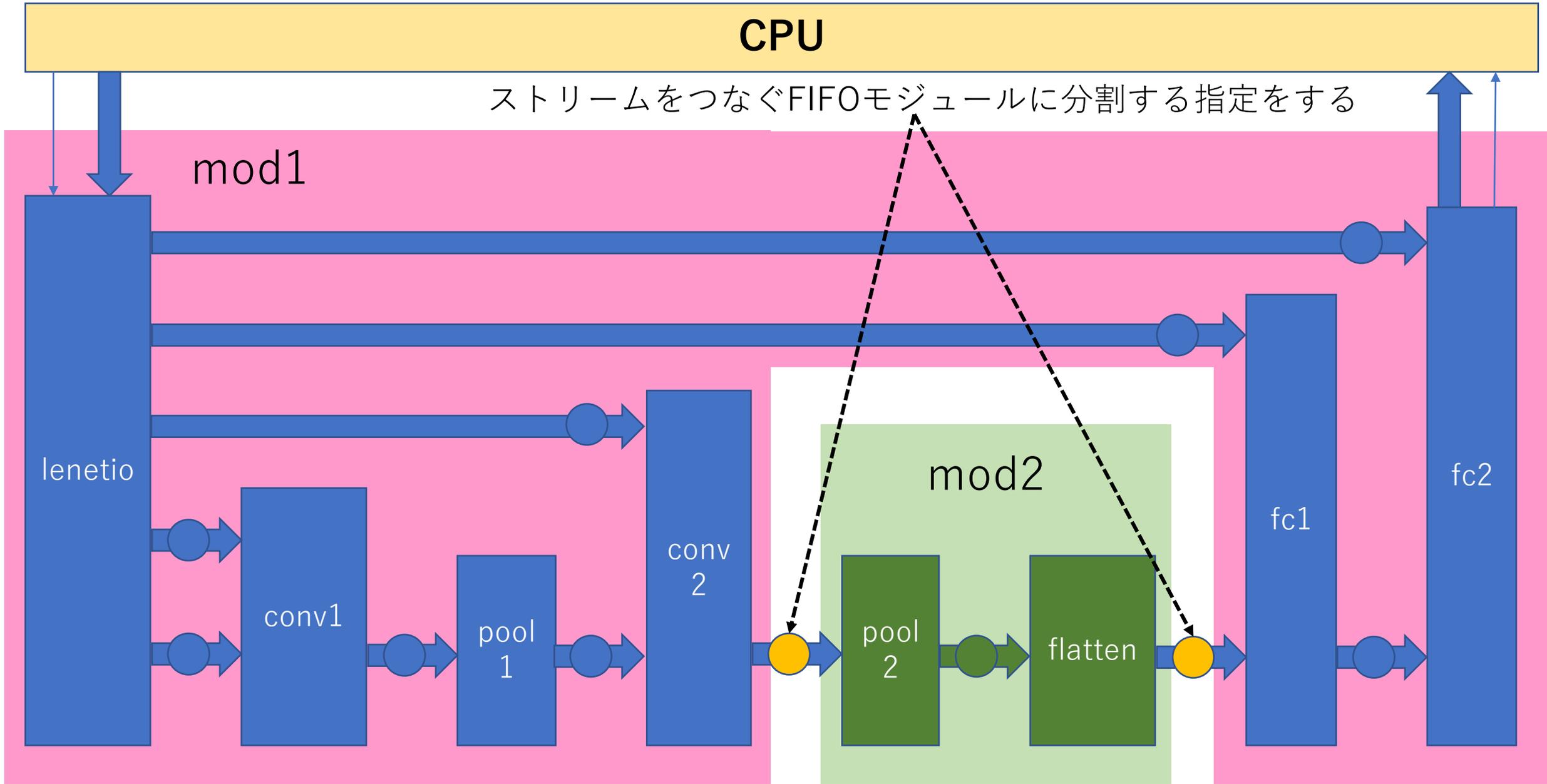
デモ

自動分割

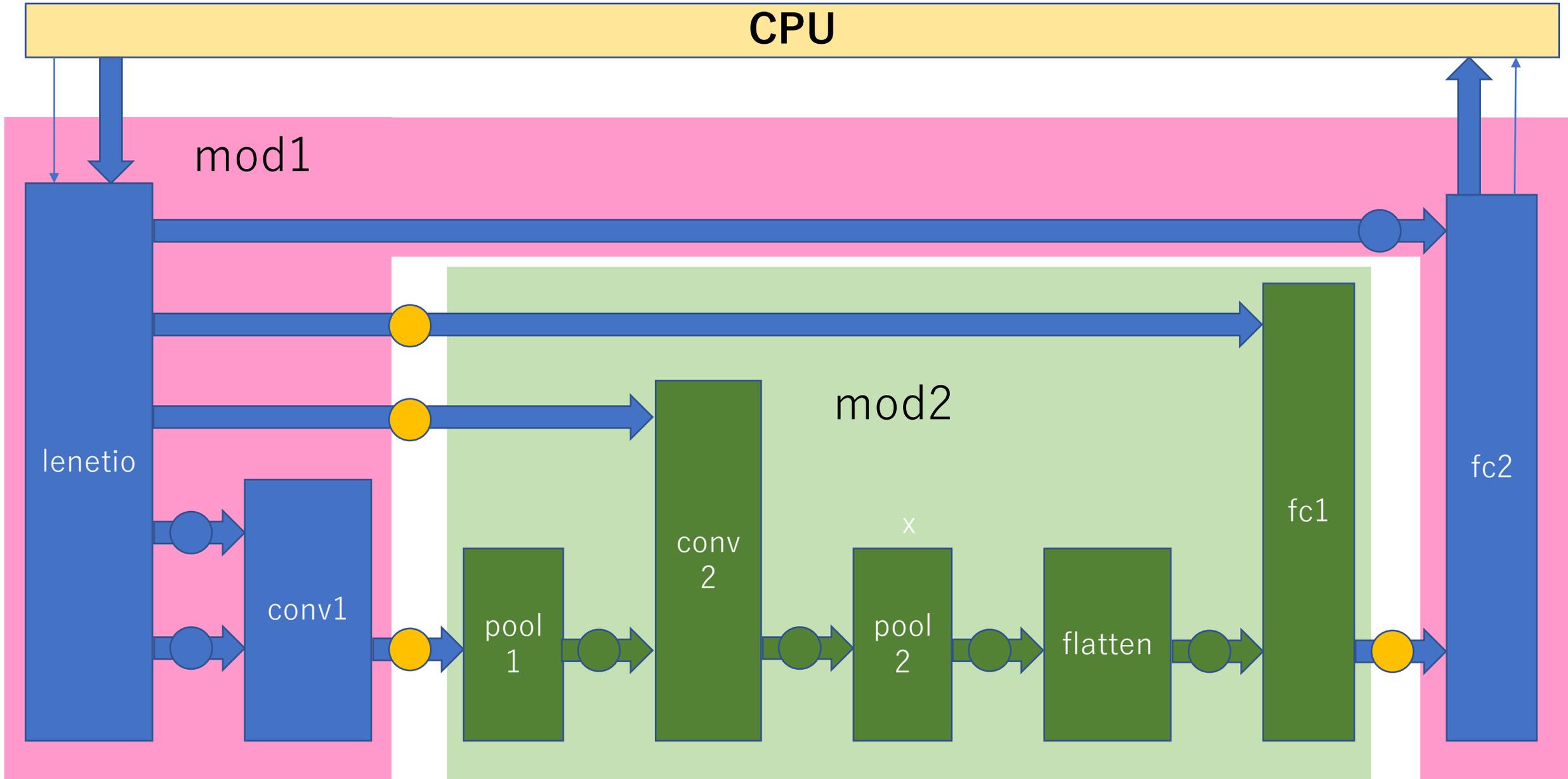
original : 分割前のモジュール構成



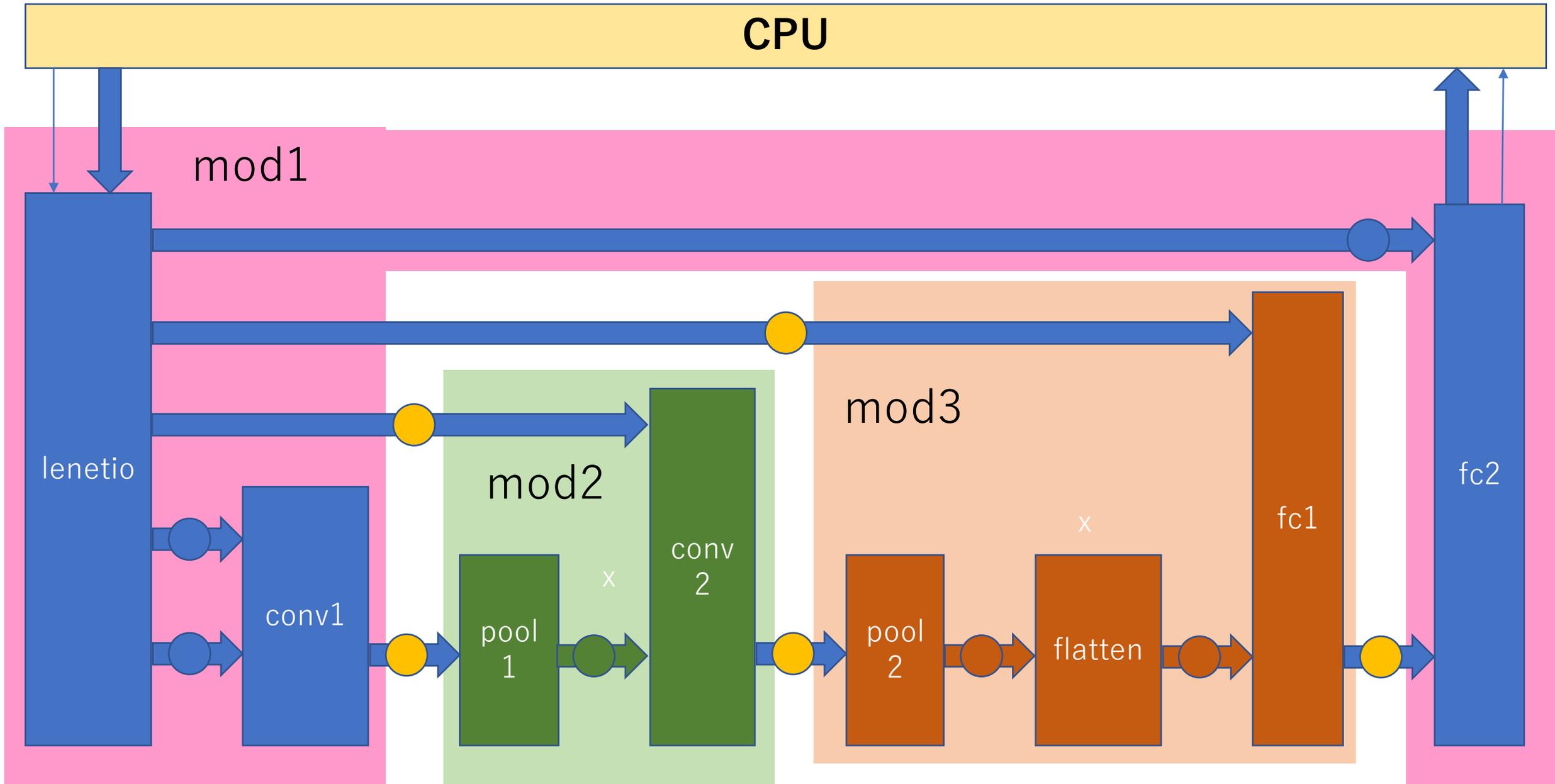
partition1 : 例1 : 2つのFPGAに分割



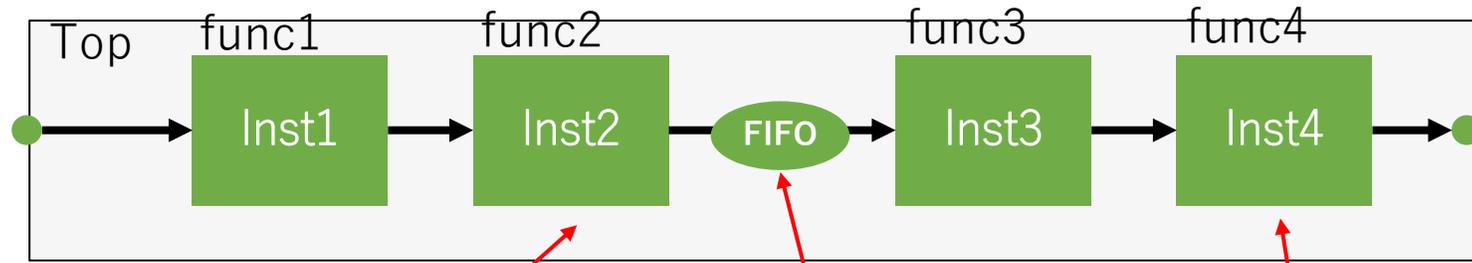
partition 2 : 例 2 : 2つのFPGAに分割



partition 3 : 例 3 : 3つのFPGAに分割



モジュール分割機能向けGUI対応 (1/2)



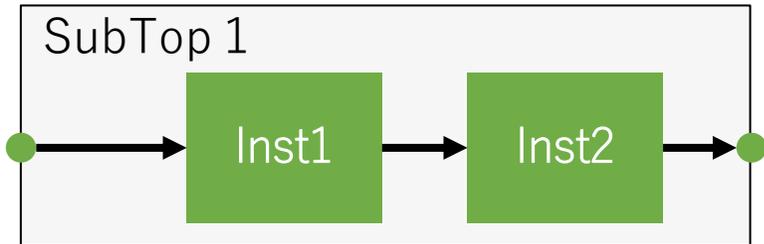
module_group = SubTop1

partition=4

module_group = SubTop2

アトリビュートで各インスタンスの分割先を指定する(将来的には別ファイルにしたい)

```
bdltrn -Zmodule_divide=SubTop1 Top.IFF ...
```

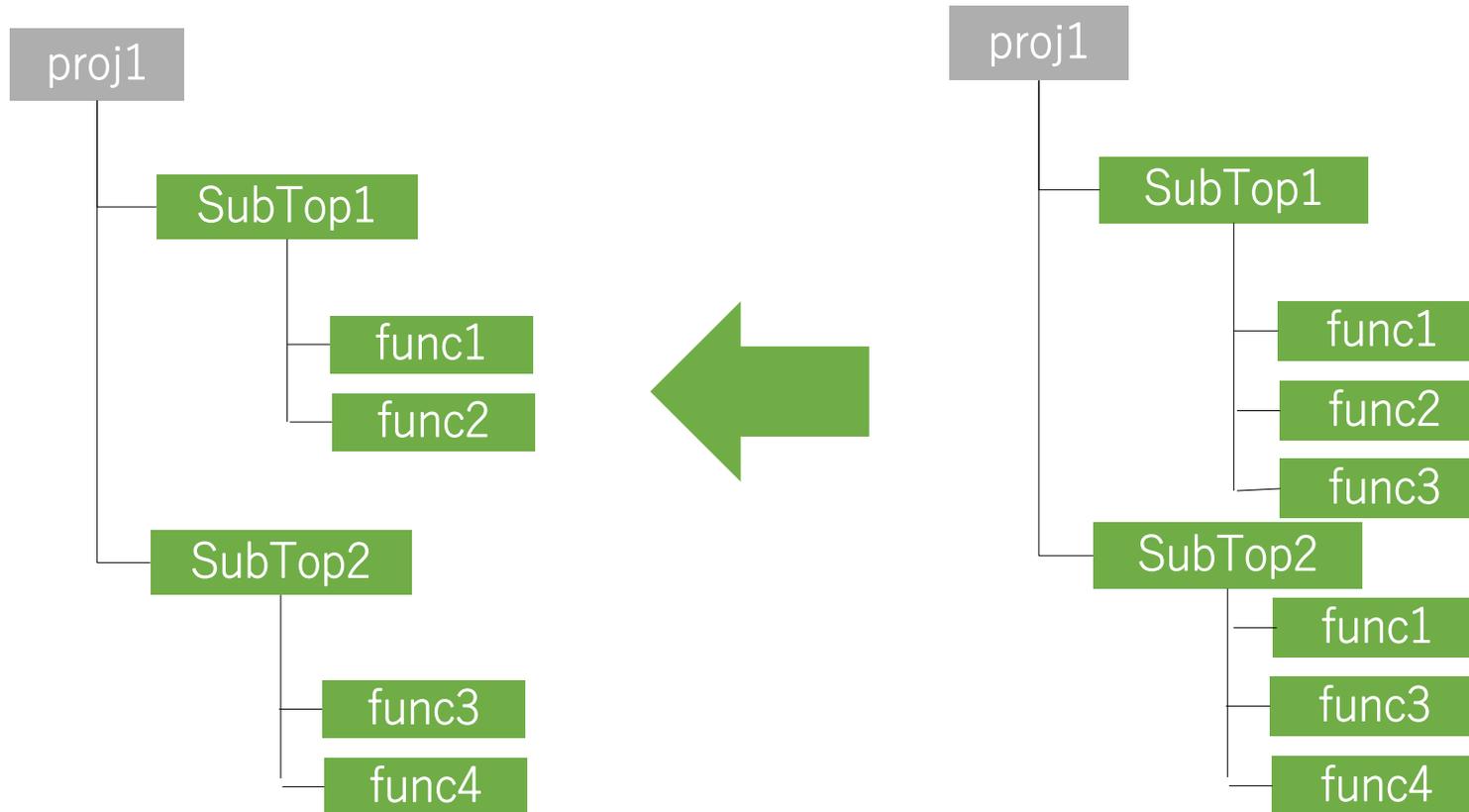


```
bdltrn -Zmodule_divide=SubTop2 Top.IFF ...
```



CWB-gui での対応

- Top.IFFの中身を見てmodule_groupアトリビュートがあればTopではなくSubTop1とSubTop2のプロセスフォルダを作る
- 下位インスタンスのプロセスフォルダに関してfunc1とfunc3はTop.IFFを見てもどのプロセスの下位になるか分からない場合はSubTop1とSubTop2の両方にいれて合成する（合成後は分かるのでfunc1はSubTop1だけに、func3はSubTop2だけにする）



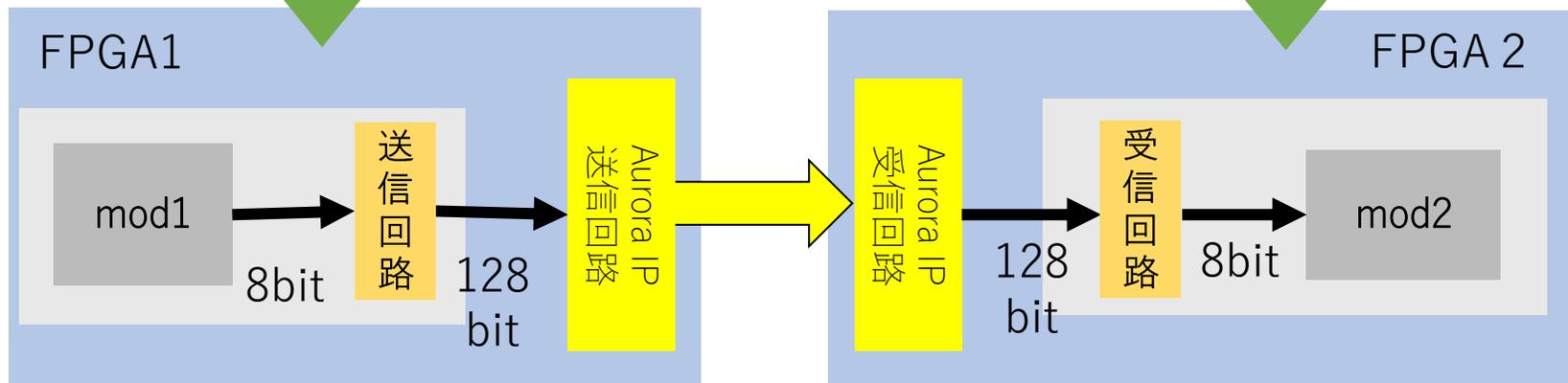
動作記述を複数FPGAで動かす設計環境(若林G)

目標：巨大なシステムを高位合成する場合に、複数FPGAに自動分割しマッピング

- M-KUBOS はFPGA間をAurora IPで通信
Aurora IP と各モジュールとのデータ通信は 170bitのAXI4-Streamで固定
- 複数のデータをパックして、170bitとして送信する回路と、複数個のデータ
複数個のデータをまとめる送信回路と**複数個のデータに分割する受信回路**が必要
- 高位合成ツールで**送信回路と受信回路を自動生成** (昨年度成果)

```
// Cyber func=process  
void mod1(sc_int<8> ary[256]  
  /* Cyber stream_pack=16 */){//16データを1パケットとせよ  
}
```

```
// Cyber func=process  
void mod2(sc_int<8> ary[256]  
  /* Cyber stream_pack=16 */){  
}
```



動作記述を複数FPGAで動かす設計環境(若林G)

- 本年度上期成果：分割指定の容易化

「**分割位置のFIFO**」と「**分割後のモジュール名**」を合成ディレクティブ指定すると、分割された高位合成用記述と、FPGAボード間の通信に必要な回路記述を自動合成

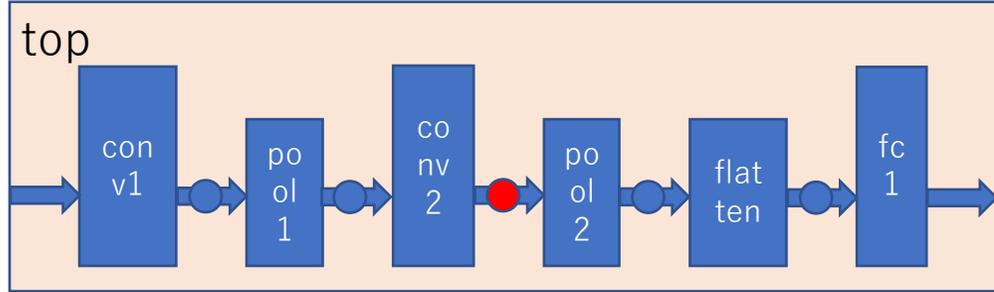
```
SC_MODULE(top){  
  ...  
  pool1 *inst_pool1;  
  conv2 *inst_conv2;  
  cw_fifo *inst_fifo3;  
  pool2 *inst_pool2;  
  ...  
}
```

**SystemC
動作記述**

分割指定なし

高位合成ツール

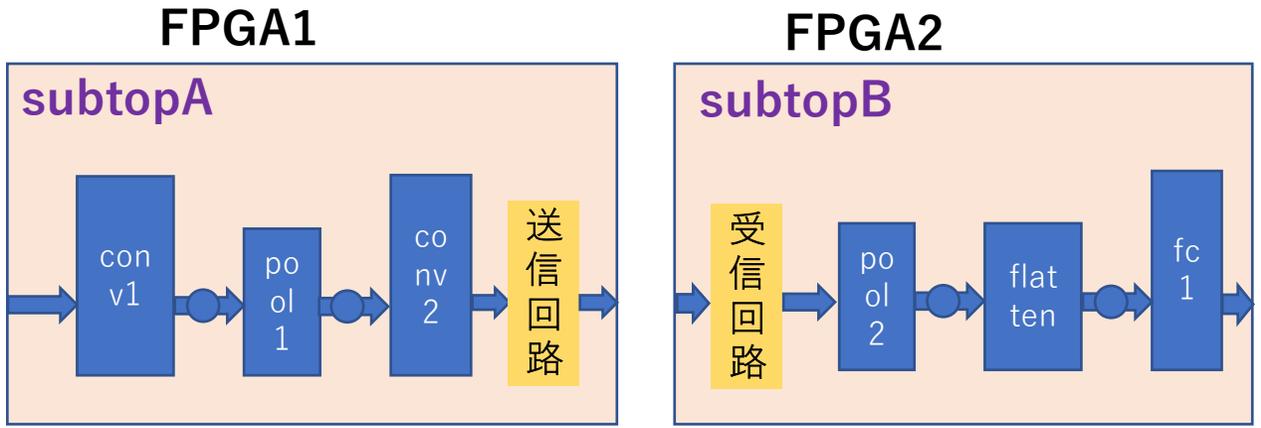
RTL記述：単一FPGA



複数FPGAへの分割指定

```
SC_MODULE(top){  
  ...  
  pool1 *inst_pool1;  
  conv2 *inst_conv2/* Cyber module_group = subtopA */;  
  cw_fifo *inst_fifo3/* Cyber partition = 16 */;  
  pool2 *inst_pool2/* Cyber module_group = subtopB */;  
  ...  
}
```

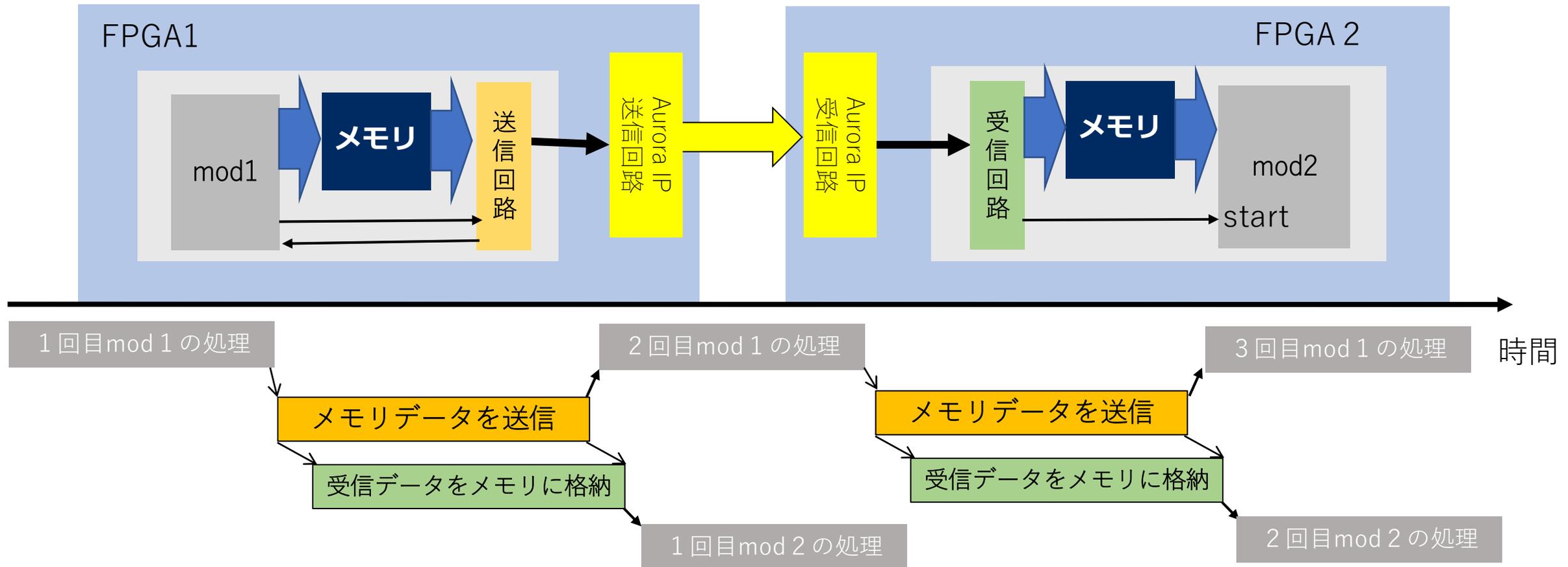
高位合成ツール



複数データを172ビット分パックして、送信、受信

動作記述を複数FPGAで動かす設計環境(若林G)

- メモリ要素をストリーム化して送信。ストリームを受信してメモリに格納するインタフェース回路を自動生成
- 上記はランダムアクセスの場合。順次アクセスの場合はメモリを使わずに、FIFOだけの回路を生成可能)



動作記述を複数FPGAで動かす設計環境(若林G)

- メモリバッファをピンポンバッファにすることで高速化 (ランダムアクセス)

