

バレルシフタとは

前回、やや消化不足の感じがするので、このままアキュムレータマシンへ爆走すると、ついていけない人、自分が何をやっているのかわからなくなる人が出るような気がする。幸いこの授業はちっとも祝日に当らず、かつ、休講の予定もないので、ここで、一回補習をしておこうと思う。

前回設計したALUはcom=100で左へinaを1ビットシフトを行ない、101で右へ1ビットシフトを行なった。このシフト機能は実際には図中の左側に示すようにデータセクタで実現される。このシフト機能は、1ビットに限定されているし、この時、inbは何の役にも立っていない。そこで、以下の機能に改造してみよう。

例: com=100で、inbの下の2ビットの示すビット数分inaを左にシフトする。つまり、inbが00ならば0bit, 01ならば1bit, 10ならば2bit, 11ならば3bitシフトする。(今回、inaは4ビットなので、これ以上シフトするとなくなってしまう)従来シフト同様、空いた桁には0を埋める。

このシフトを行なうためのシフタ bsl(barrel shift left)を設計してみる。最も簡単には、下のようになる。

```
module bsl{
input ina<4>, inb<2> ;
output y<4> ;
instrin enable ;
instruct enable alt {
  inb==01: y = ina<2:0>||0b0 ; /* 1bit Left shift */
  inb==10: y = ina<1:0>||0b00; /* 2bit Left shift */
  inb==11: y = ina<0>||0b000; /* 3bit Left shift */
  else: y = ina;
}
}
```

この方法は高速かつ桁数が少なればうまく行くので、今回はOKだが、桁数が多いと記述も大変でハードウェア量も膨大になる。そこで、図の右の構成を考える。

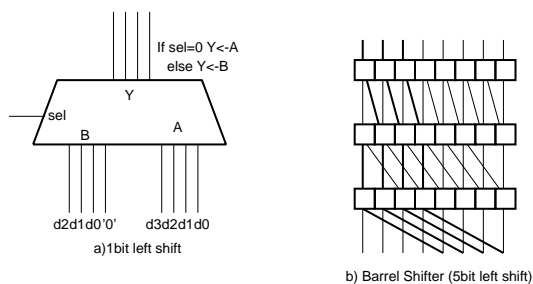


図 1: バレルシフタの構成

この方式はバレルシフタと呼ばれ、 2^n ビットシフトしたデータを作るスイッチの縦列接続によって、任意のデータのシフトを実現する方式で、一種のMIN(Multistage Interconnection Network)と考えることもできる。sf1で記述すると下のようになる。

```

module bsl{
input ina<4>, inb<2> ;
output y<4> ;
instrin enable ;
sel_v aa<4>;
instruct enable par {
alt {
    inb<0>: aa = ina<2:0>||0b0 ; /* 1bit Left shift */
    else: aa = ina;}
alt {
    inb<1>: y = aa<1:0>||0b00; /* 2bit Left shift */
    else: y = aa;}
}
}

```

演習1: このバレルシフタを alu4.sfl に組み込み動作を確認せよ。

演習2: 上記の記述を参考に右バレルシフタ (bsr) を構成し、com=101 で左シフト同様に動作するように組み込め。