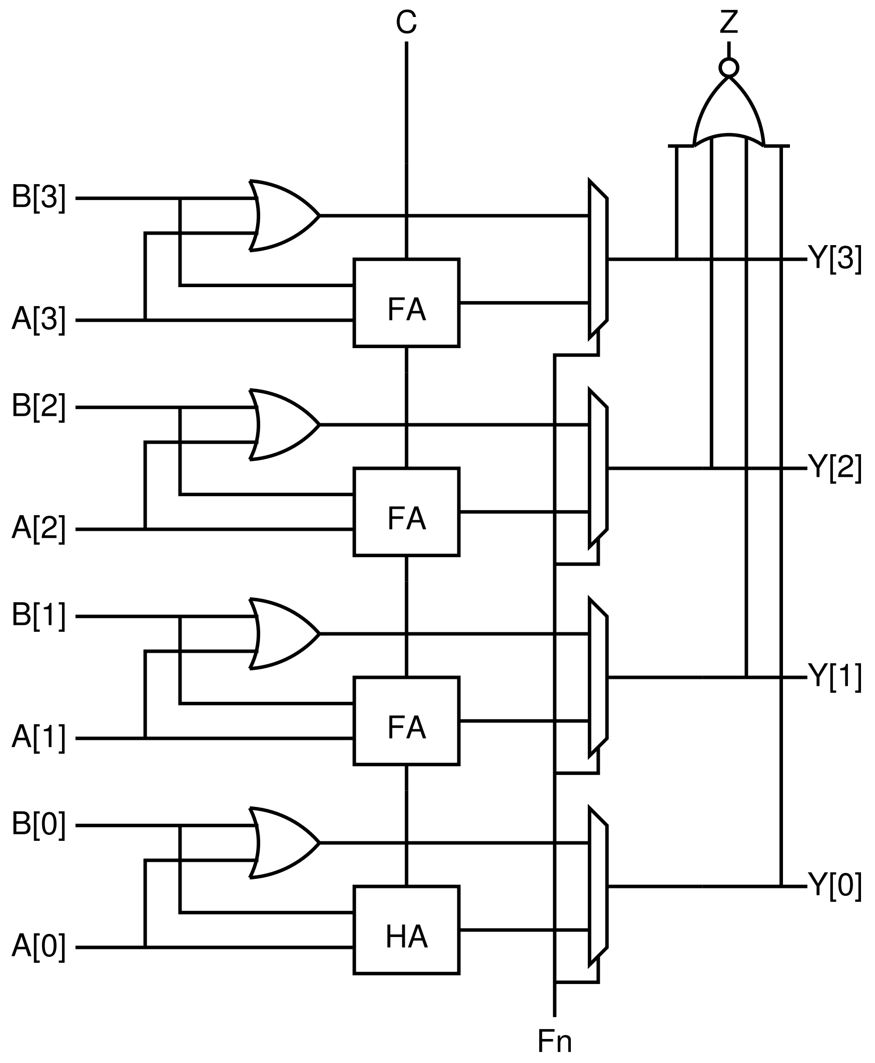
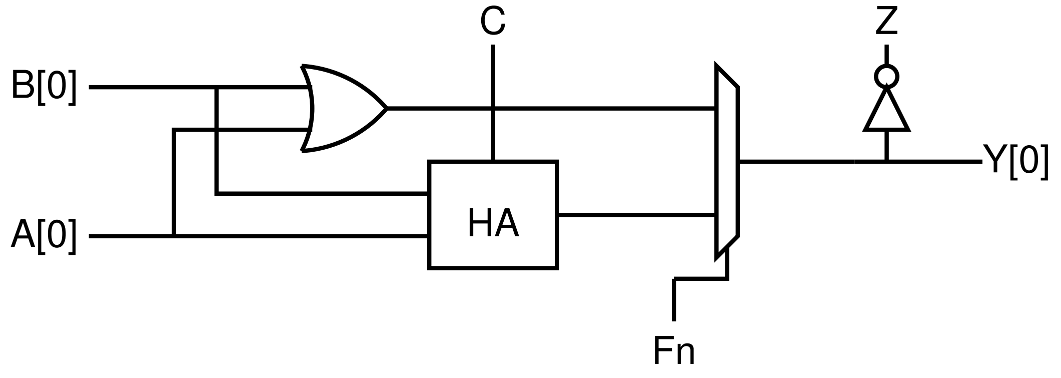
Bit Slicing

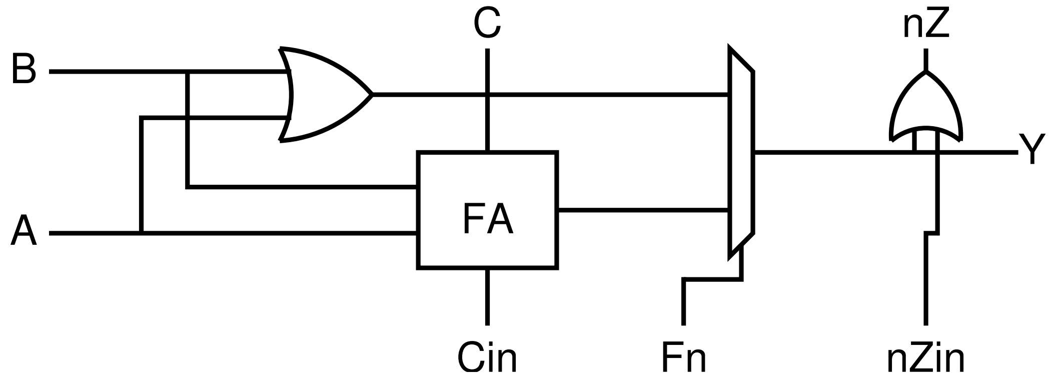


Bit Slicing

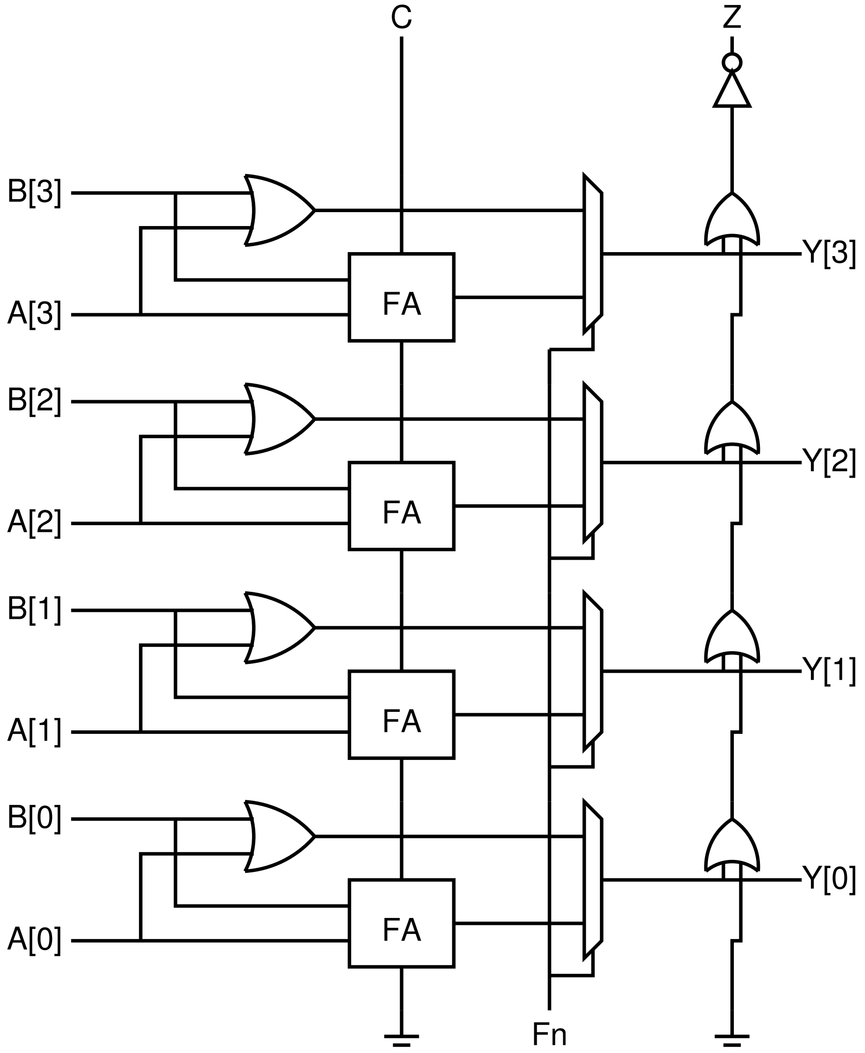
1 Bit ALU (no consideration of bitslicing)



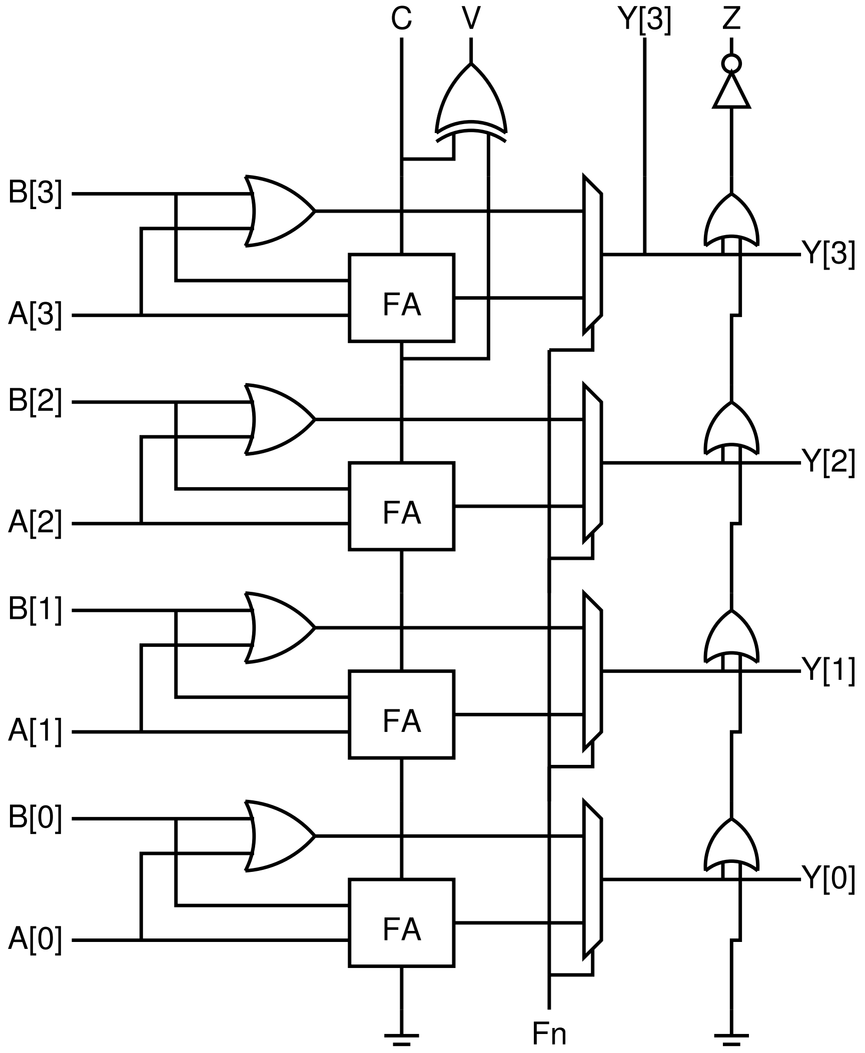
Bitslice Design for ALU



Bit Slicing



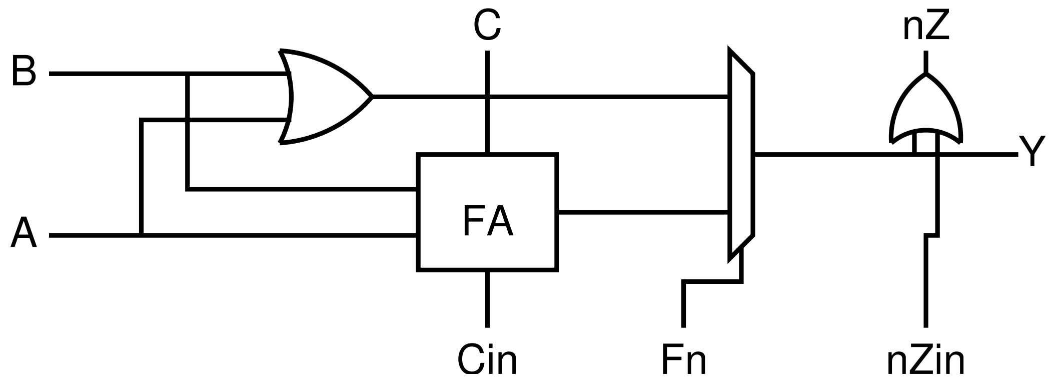
Bit Slicing



2004

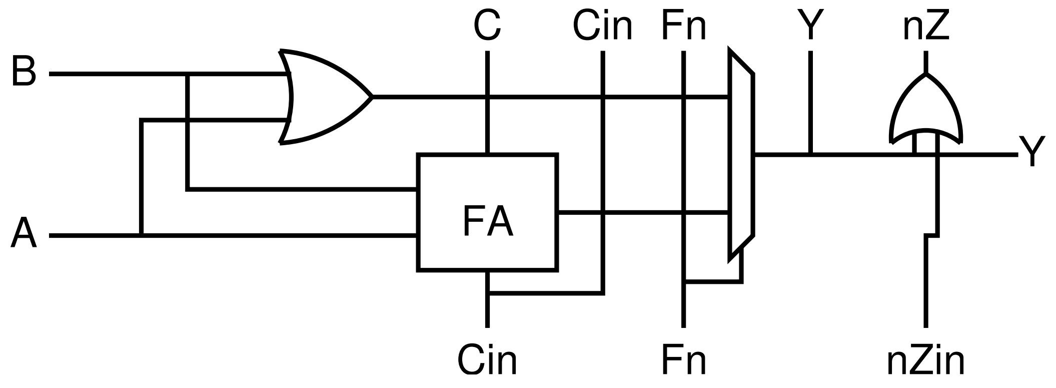
Bit Slicing

Original Bitslice Design for ALU

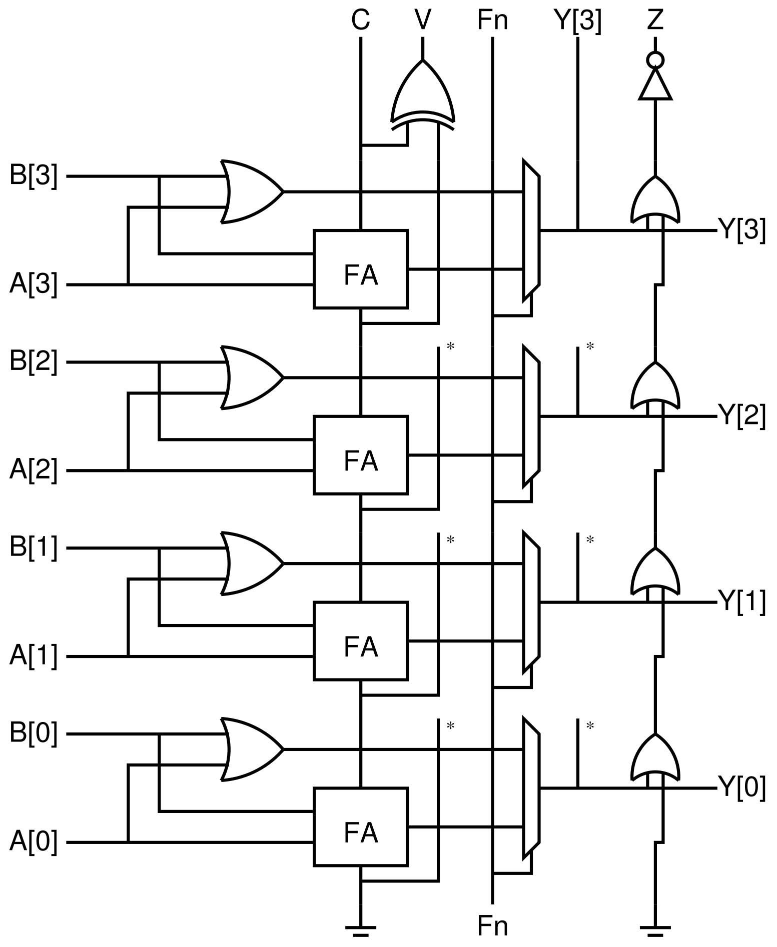


Modified Bitslice Design for ALU

- Wiring is arranged so that no over cell routing is required.



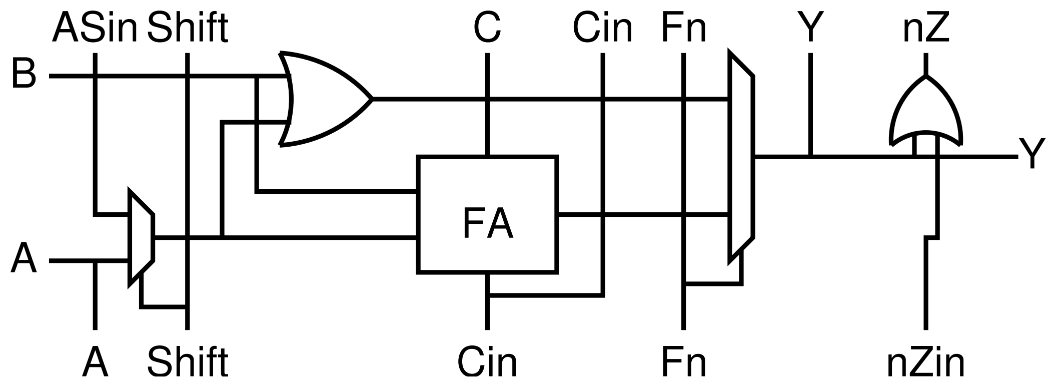
Bit Slicing



*a number of wires are left unconnected

Bit Slicing

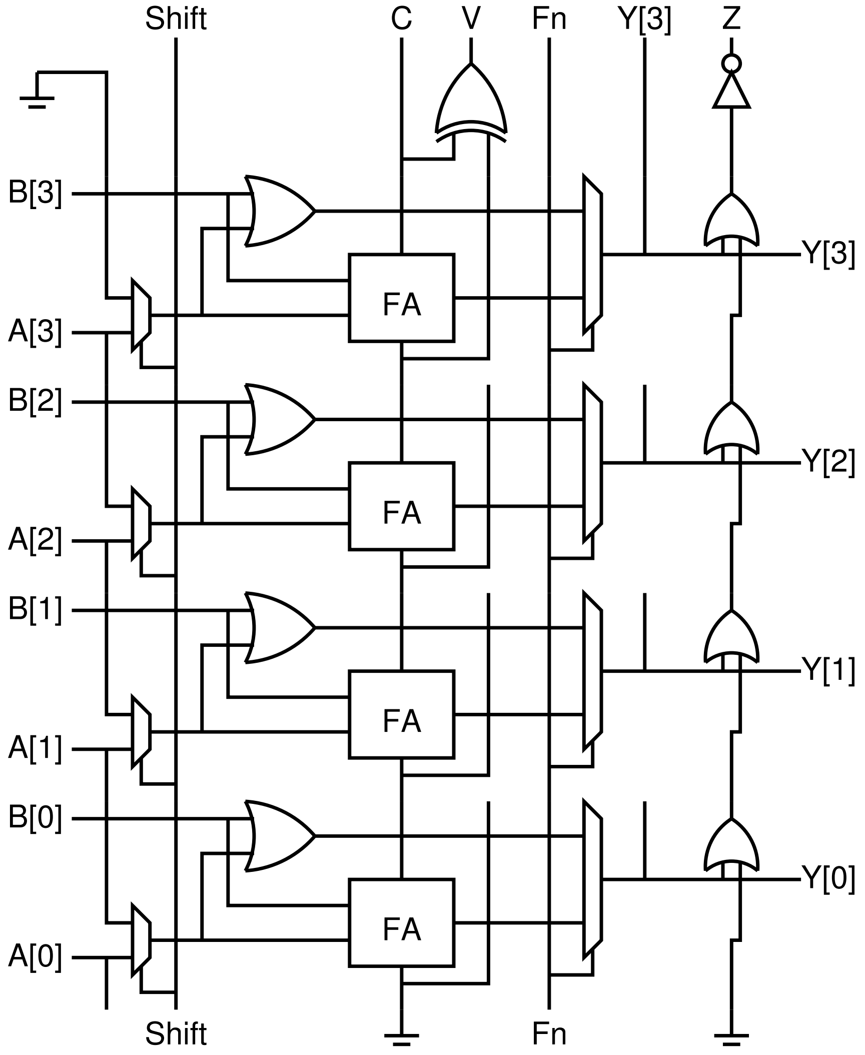
Single Bit Shift



This simple ALU bitslice now supports 4 functions

- `Y = A + B`
- `Y = A | B`
- `Y = (A >> 1) + B`
- `Y = (A >> 1) | B`

Bit Slicing

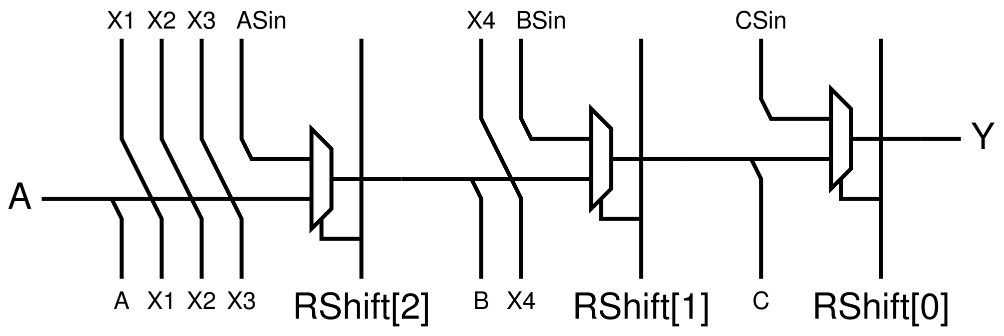


2008

Bit Slicing

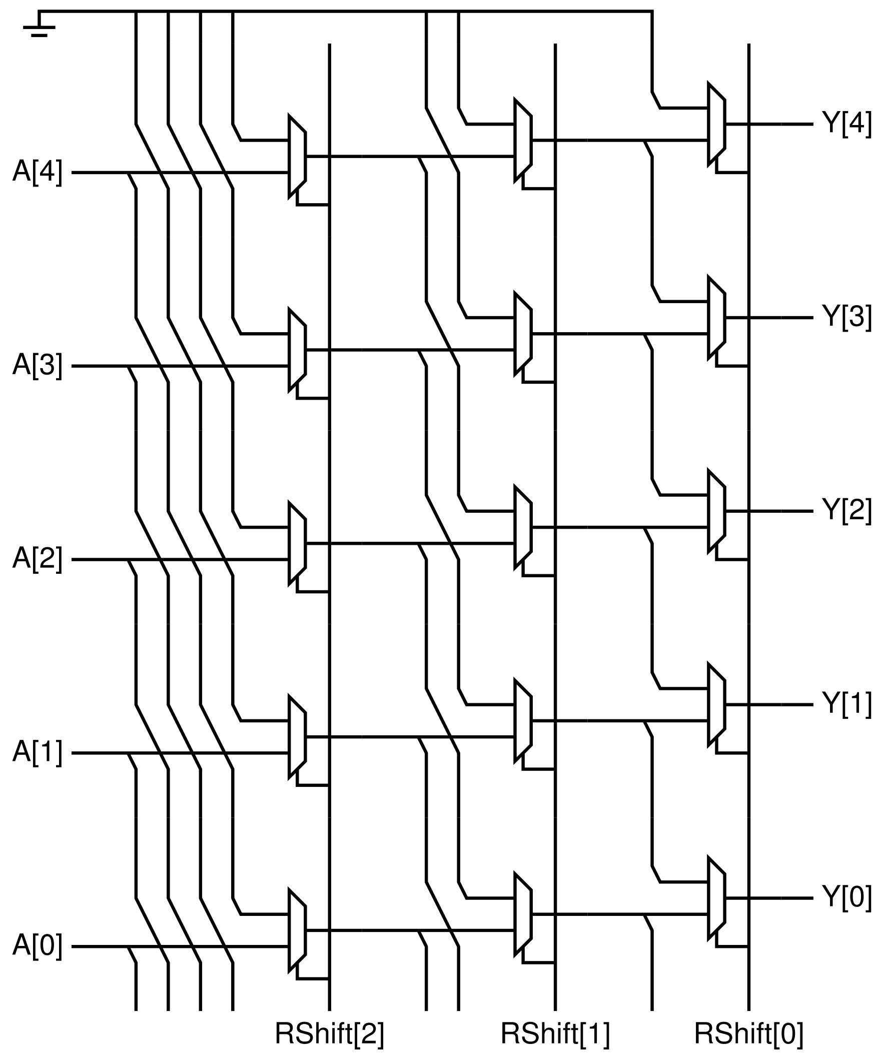
Multi-Bit Shift

This barrel shifter bitslice includes wiring for 4 bit right shift, wiring for 2 bit right shift and wiring for 1 bit right shift.



- `Y = A >> RShift[2:0]`

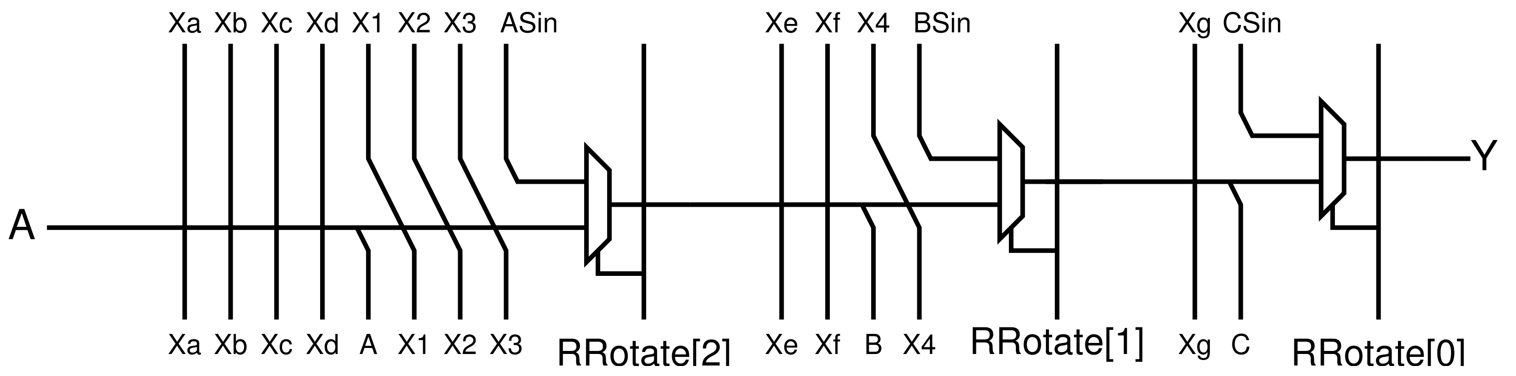
Bit Slicing



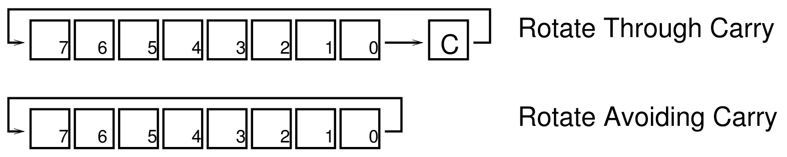
Bit Slicing

Multi-Bit Rotate

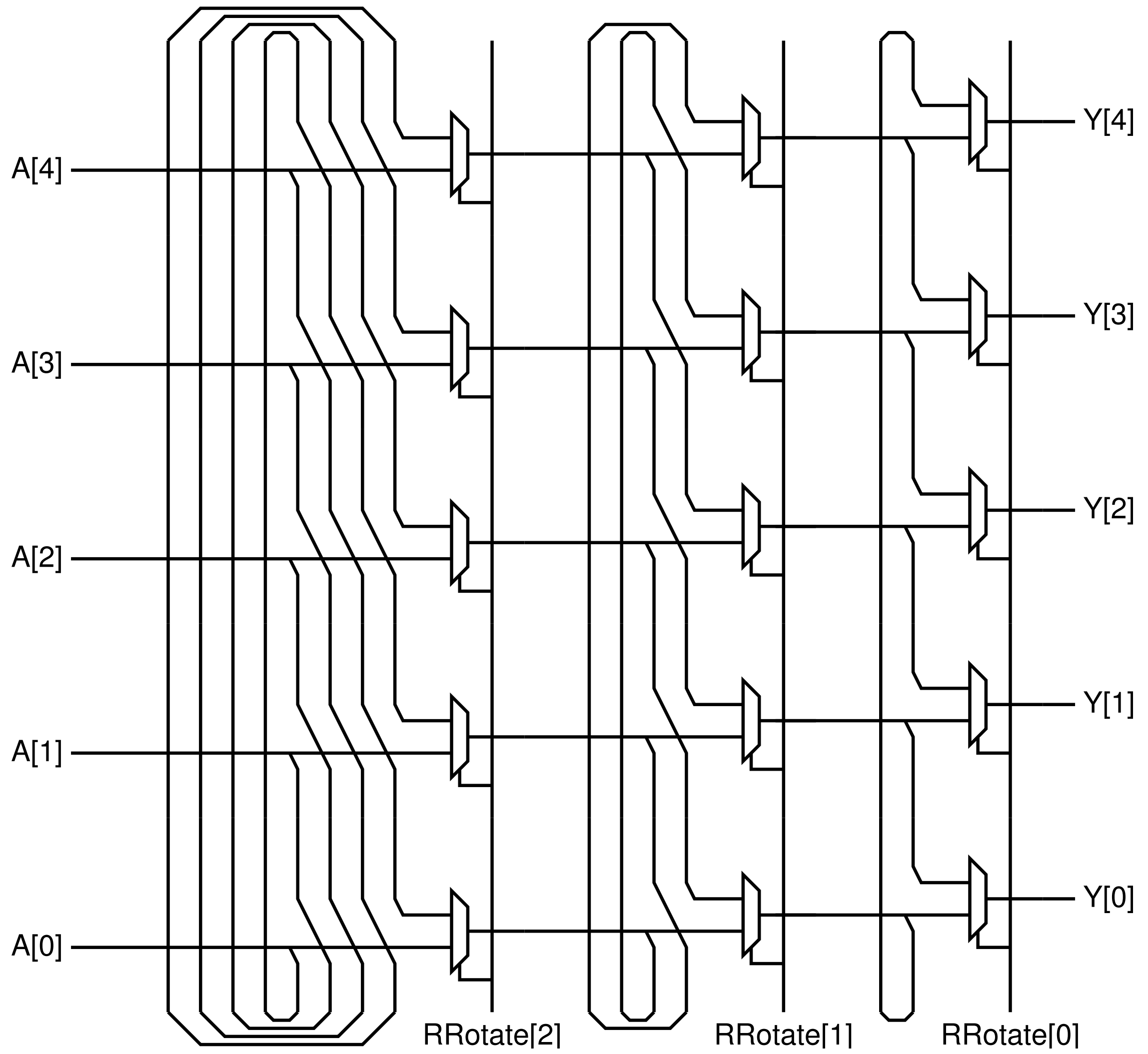
By providing extra feedthrough paths we can create a bitslice for multi-bit rotation.



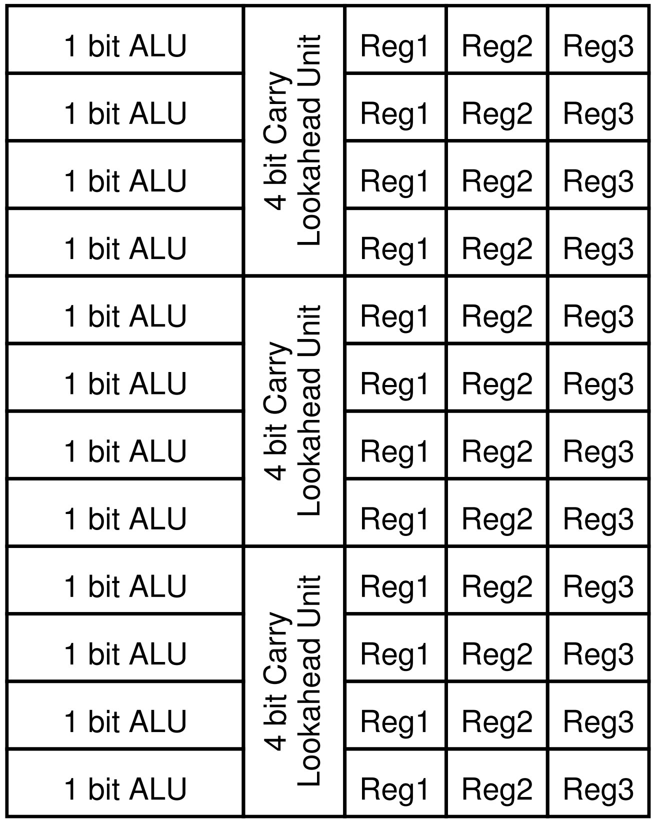
N.B. Most single bit rotate functions will *rotate through the carry*. This barrel rotation, based on the SPARC implementation, does not make use of the carry for input or output.

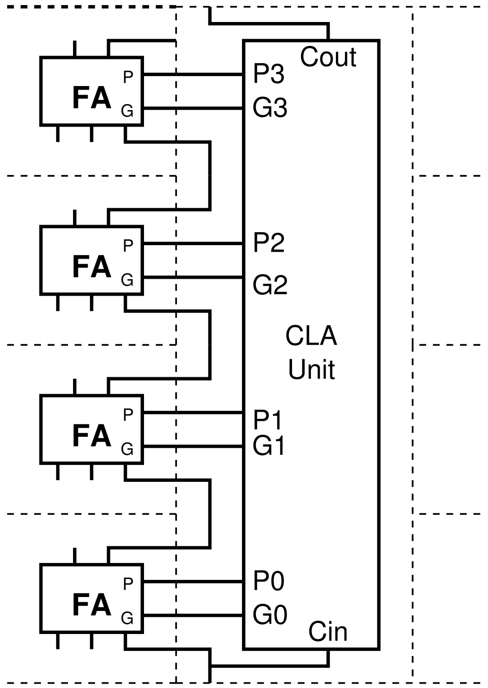


Bit Slicing



Bit Slicing





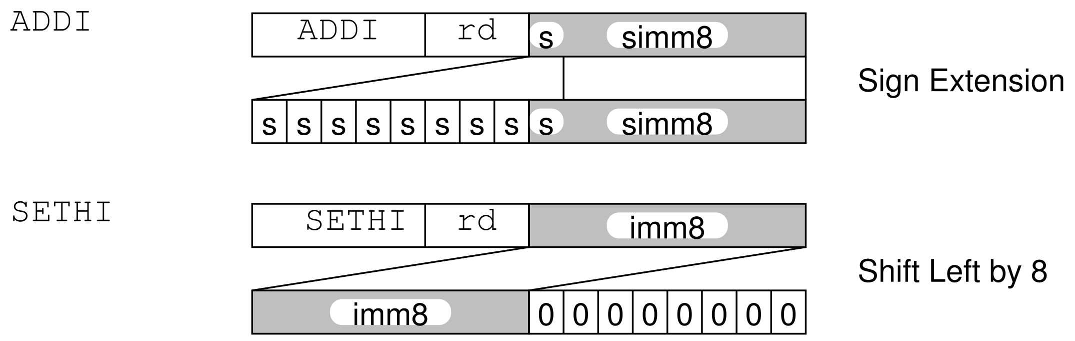
- Bitslice Exceptions

  Where full bitslicing is not suitable we attempt to disrupt the bitslice as little as possible.

Bit Slicing

Immediate Alignment

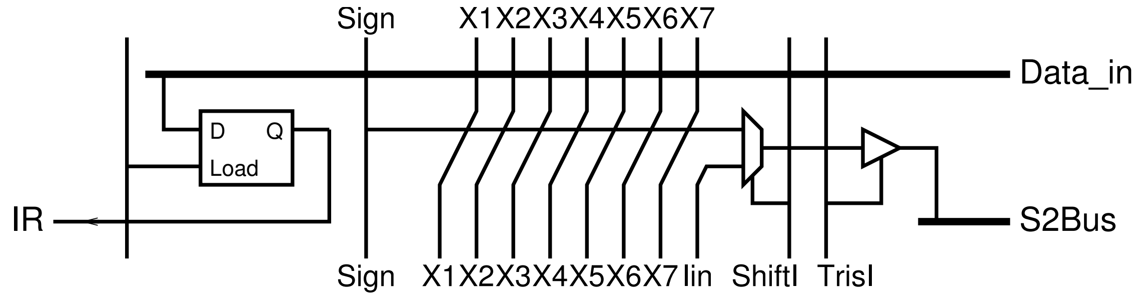
Here an 8 bit data field within a 16 bit instruction may be interpreted as either an 8 bit signed immediate (in need of sign extension) for an ADDI instruction or as the most significant 8 bits of an immediate value (in need of 8 bit shift left) for a SETHI instruction.



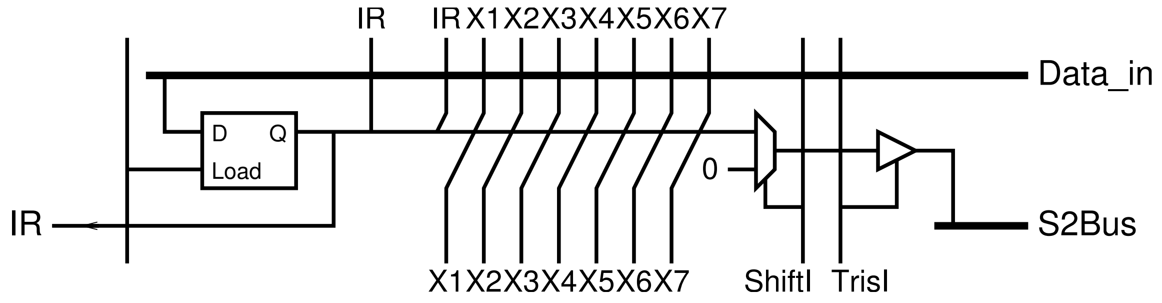
```
ADDI  simm8,Rd                 Rd <- Rd + simm8
SETHI  imm8,Rd                 Rd[15:8] <- imm8   Rd[7:0] <- 0
```

Bit Slicing

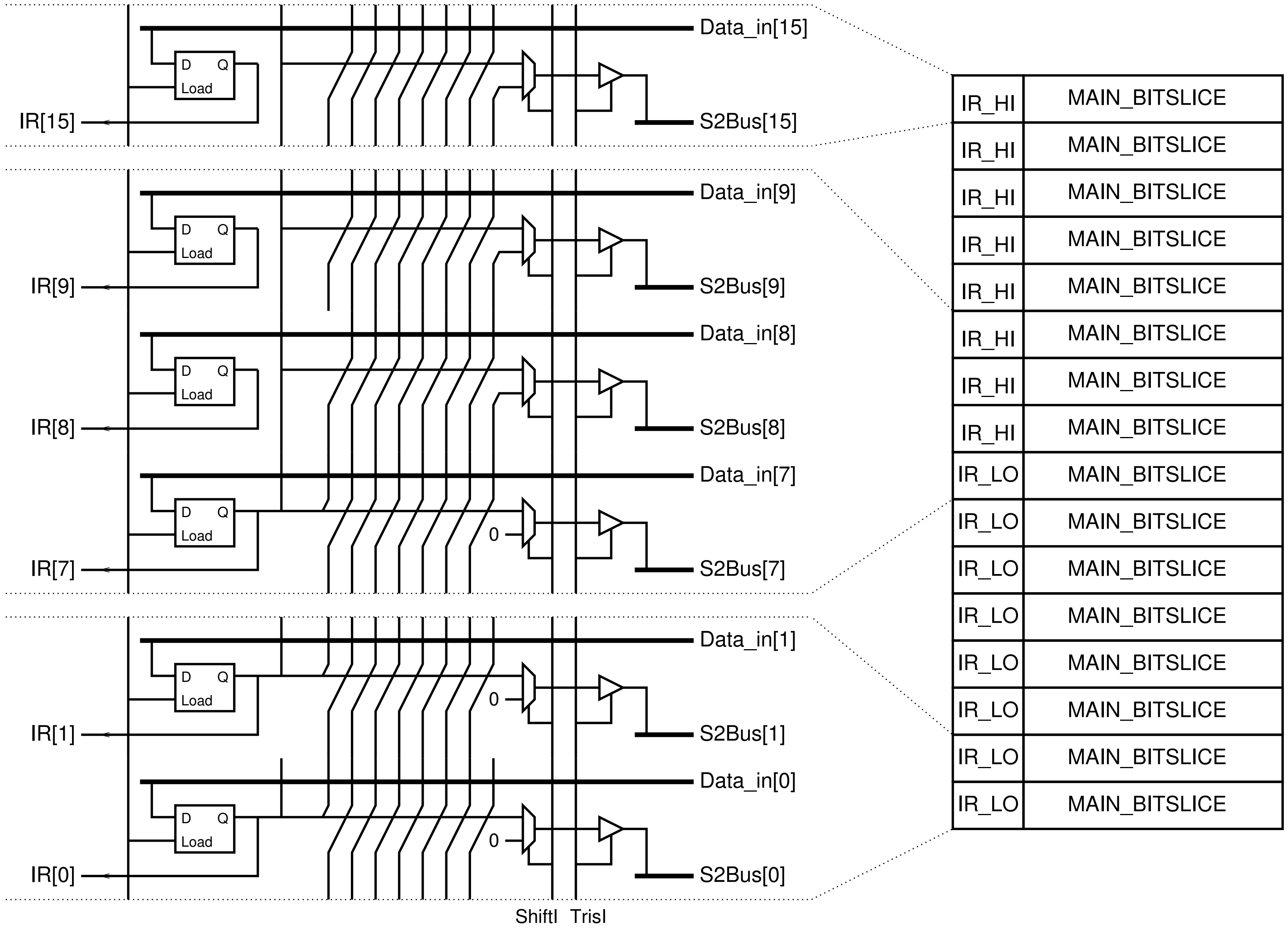
- **IR_HI** bitslice is used for bits 8–15



- **IR_LO** bitslice is used for bits 0–7



Both bitslices include wiring for 8 bit shift left. **IR_LO** feeds values into the shifter while **IR_HI** takes values from the shifter.


Data_in[15]
IR[15]
S2Bus[15]
D
Q
Load
Data_in[9]
IR[9]
S2Bus[9]
Data_in[8]
IR[8]
S2Bus[8]
Data_in[7]
IR[7]
S2Bus[7]
0
Data_in[1]
IR[1]
S2Bus[1]
Data_in[0]
IR[0]
S2Bus[0]
ShiftI
TrisI
IR_HI
IR_LO
MAIN_BITSLICE