Arithmetic Logic Unit

The Arithmetic Logic Unit (ALU) of the CGRA can take a number of inputs (by default configured to 4 inputs) and perform arithmetic and logic operations on two of these inputs. The inputs are specified by the instruction and selected by the two 4-input multiplexers in the top of the figure below.

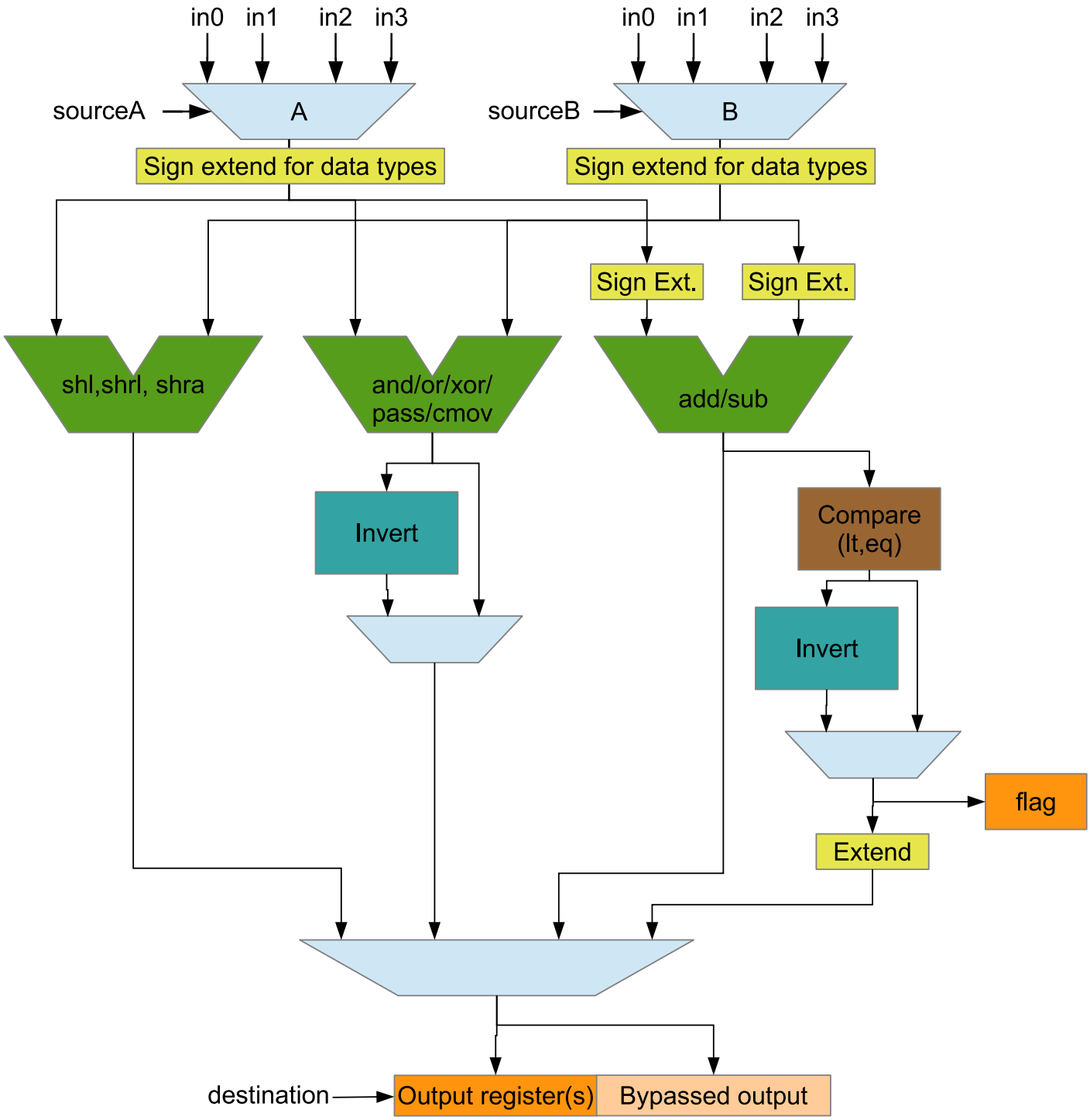
The operations the ALU can perform are divided over three functional groups:

- Shift operations
- Logic Operations
- Arithmetic operations

Additionally, the output of the arithmetic operations can be used for comparing two operands. The result of the comparison is stored in the ‘flag’ register and can be used for CMOV operations. The comparison output can also be routed to the data path, this allows transmitting the flag to other ALUs in the pipeline or using it as a a result (e.g. binarization). When routed to the data path the flag is extended from 1-bit to the width of the data path. This means that flag=0 will result in a value of 0 on the data path while flag=1 will result in 2^D_WIDTH-1. The flag can also be inverted such that not only LT (Less Than) and EQ (Equals) are available but NEQ (Not Equal) and GE (Greater than or Equal) as well.

The output of the logic module can be inverted which results in the operations NAND, NOR, XNOR and Invert.

The destination operand specifies to which register the output has to be written. Output 0 is an buffered or unbuffered output (depending on configurationBit[0], 0=unbuffered and 1=buffered), whereas the other outputs are buffered. The unbuffered output could be used to construct single-cycle complex operations.


in0
in1
in2
in3
sourceA
A
Sign extend for data types
in0
in1
in2
in3
sourceB
B
Sign extend for data types
Sign Ext.
Sign Ext.
shl,shrl, shra
and/or/xor/
pass/cmov
add/sub
Invert
Compare
(lt,eq)
Invert
flag
Extend
destination
Output register(s)
Bypassed output