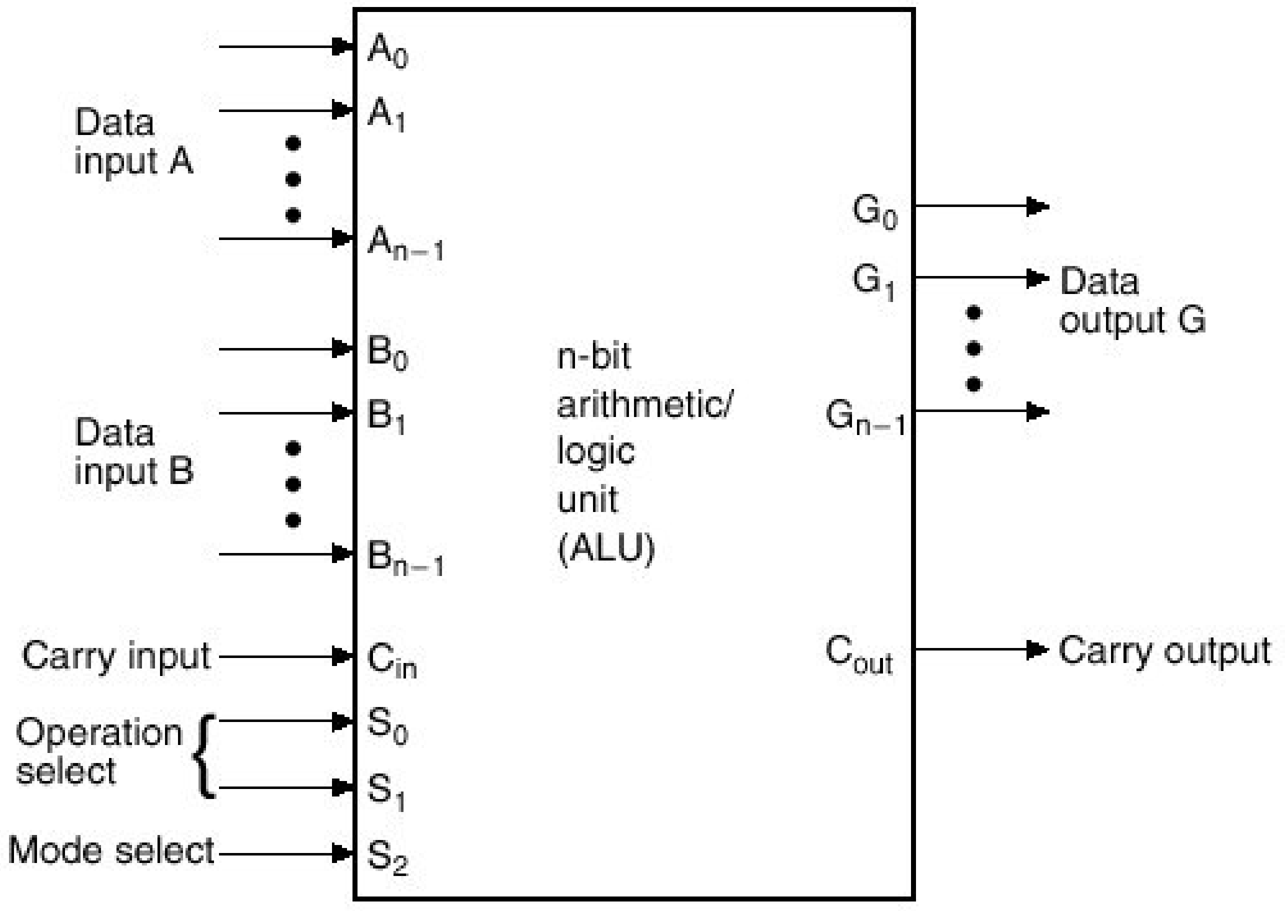
Arithmetic Circuit

- The arithmetic circuit may implemented with the following components:
  - Parallel Adder
    - Build from a cascade of full-adder circuits
  - The data input to the parallel adder is manipulated in order to achieve a number of arithmetic operations

N-bit Arithmetic/Logic Unit(ALU)

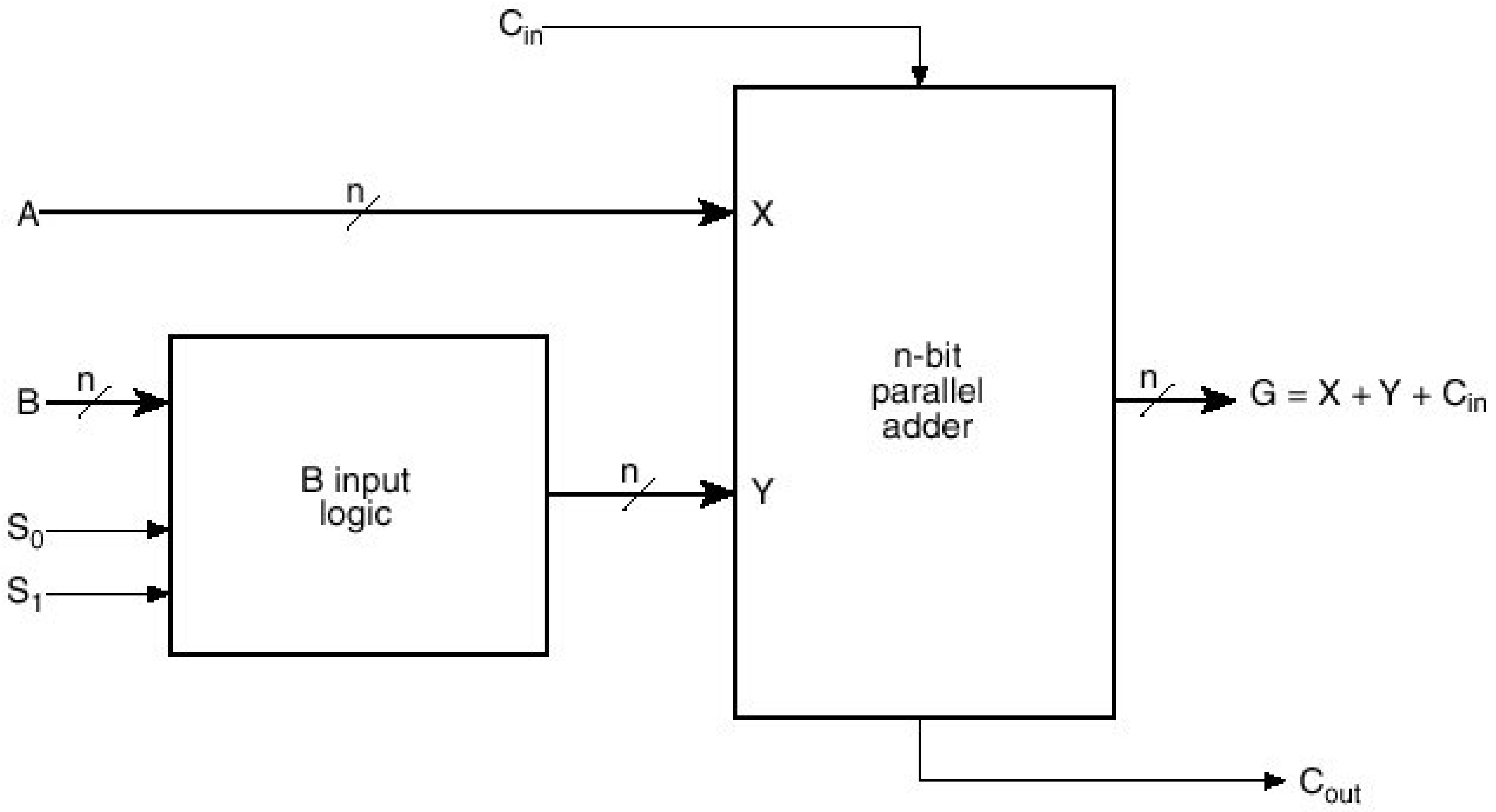


$G = A + Y + C_{in}$

- The arithmetic micro-ops can be implemented using the carry-in C_{in} and two select inputs S_1 & S_0, which condition the B input to deliver **Y** to the full-adder computing: **$G = A + Y + C_{in}$**.

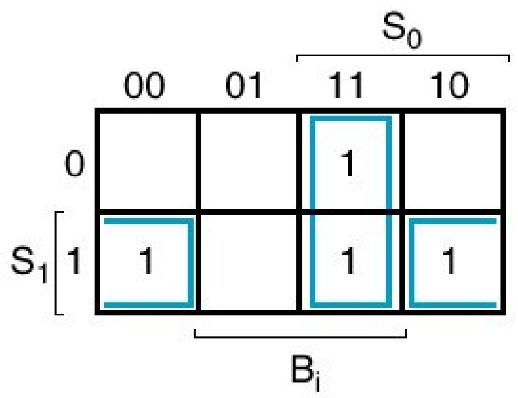
Select		Input		
S_1	S_0	Y	$C_{in}=0$	$C_{in}=1$
0	0	all 0’s	G=A	G=A+1
0	1	B	G=A+B	G=A+B+1
1	0	$\overline{B}$	G=A+$\overline{B}$	G=A+$\overline{B}$+1
1	1	all 1’s	G=A-1	G=A

Arithmetic Circuit



Y(S,B)

- The logic function **Y(S,B)** is derived as:



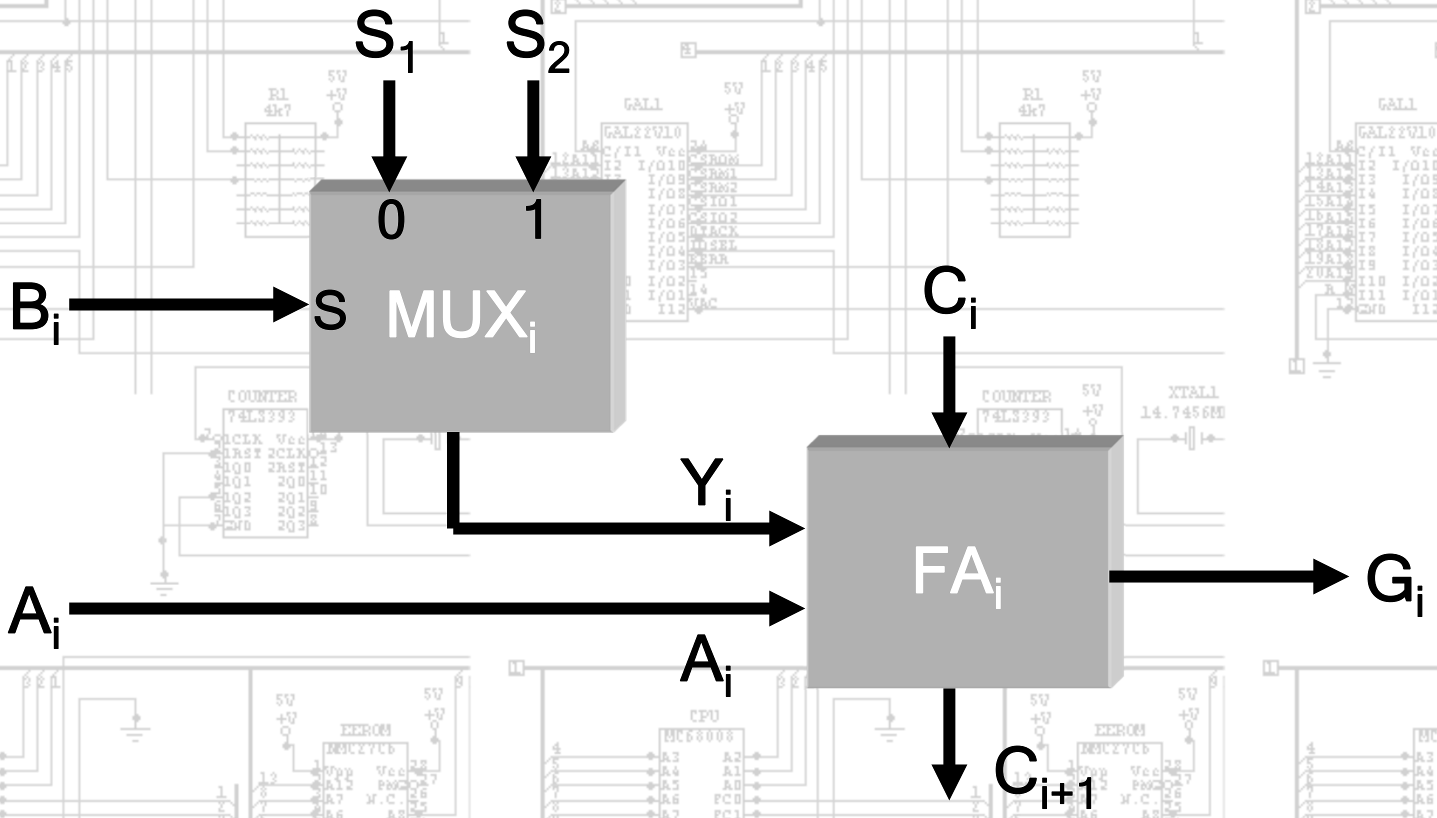
S_1 \ $S_0 B_i$	00	01	11	10
0			1	
1	1		1	1

$$Y_i = S_0 B_i + S_1 \overline{B}_i$$

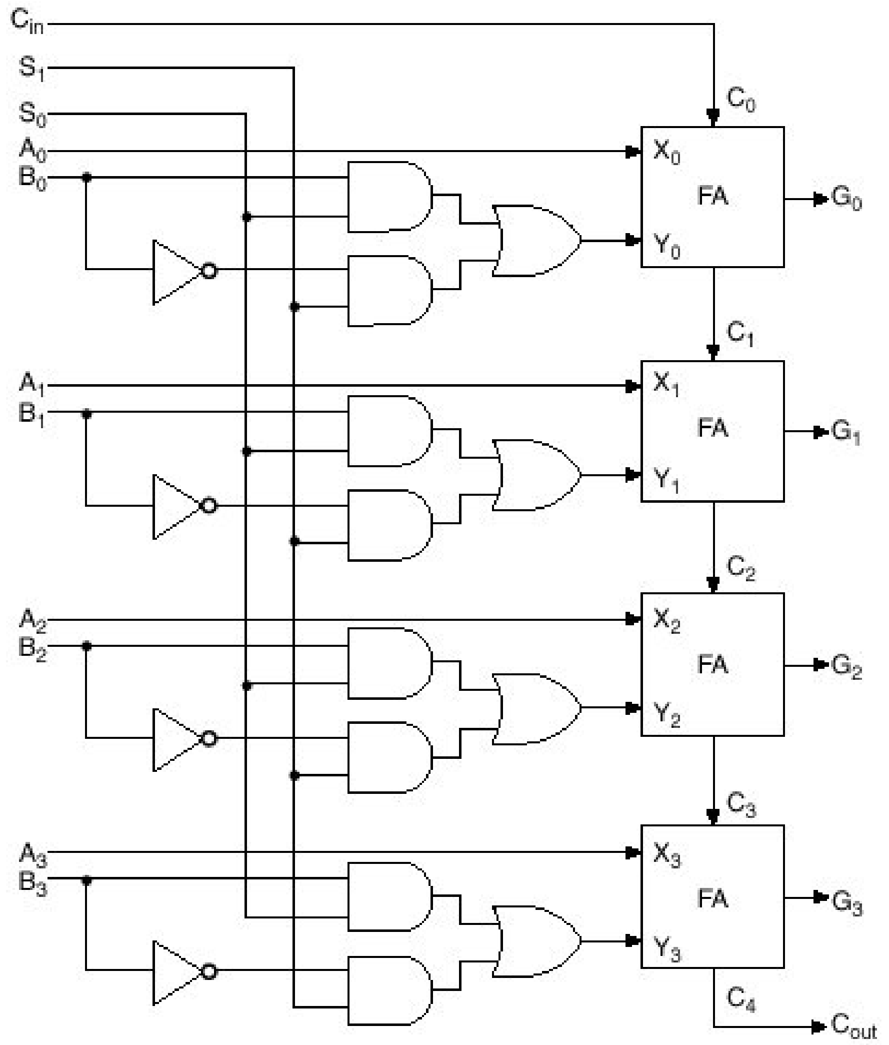
- Thus a 2:1 MUX controlled by B_i can efficiently generates Y_i

One Bit Slice

- One bit slice of the Arithmetic unit on the next slide.



4-Bit Arithmetic Circuit

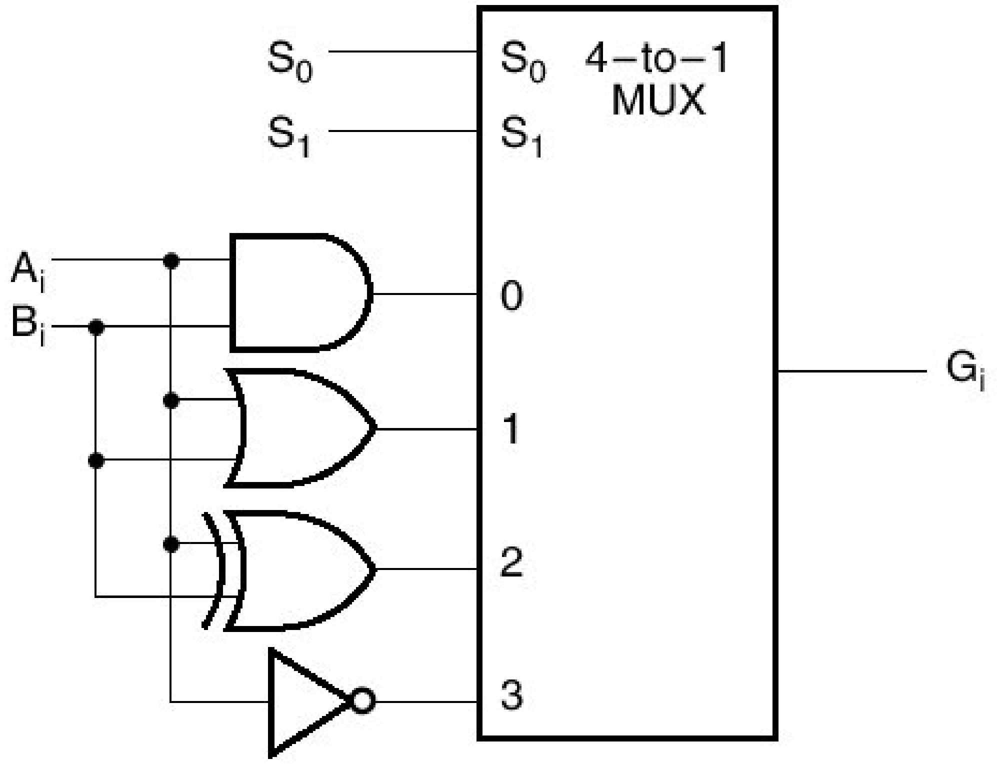


Logic Circuit

- The logic function are similarly selected by input S_1 and S_0 :

Select		Output	
S_1	S_0		
0	0	$G = A \wedge B$	AND
0	1	$G = A \vee B$	OR
1	0	$G = \underline{A} \oplus B$	XOR
1	1	$G = A$	NOT

Logic Circuit Implemented with a 4:1 MUX

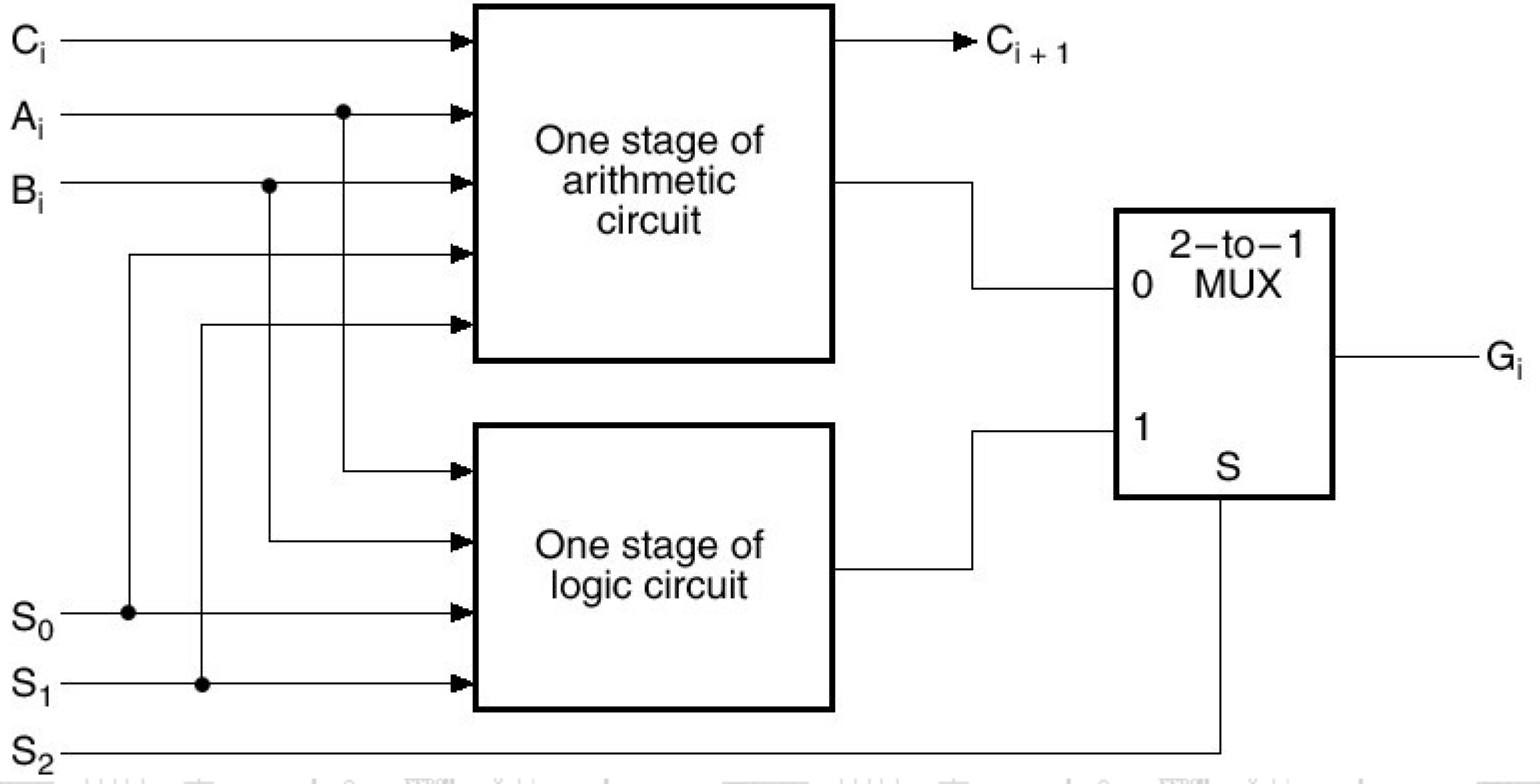


- One bit slice of the logic unit.

ALU (Arithmetic/Logic)

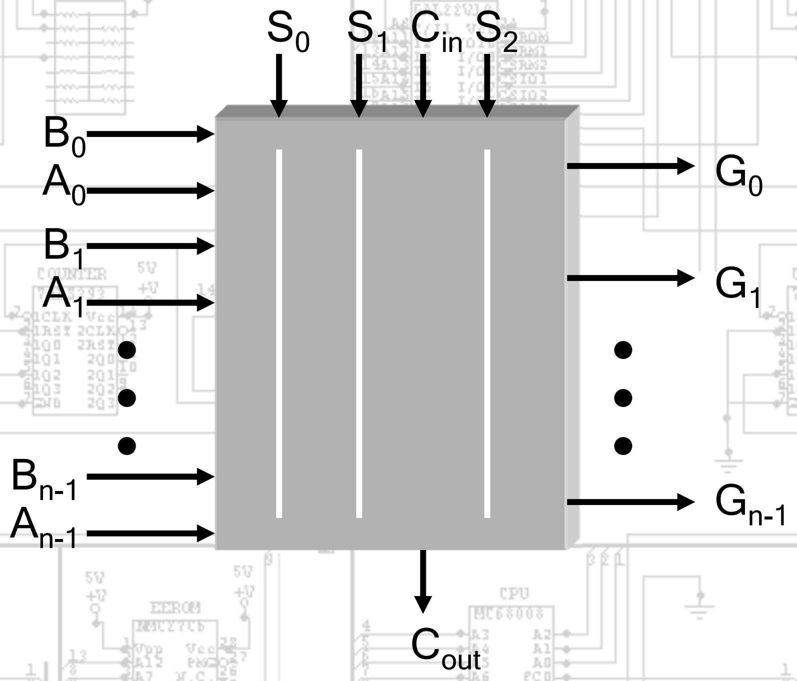
- We next use an additional 2:1 MUX controlled by S_2 to select either the arithmetic output bit or the logic output bit as shown on the next slide.

One bit slice ALU



N-bit ALU

- To construct an n-bit ALU we concatenate n-bit slices together:



Physical Implementation

- Physical schematic of an n-bit ALU assembled from a bit slices as shown on the previous slide.

1. Note the control signals, because they apply to the whole word, tend to cross the datapath.
2. This geometry results in very efficient VLSI chip implementation.

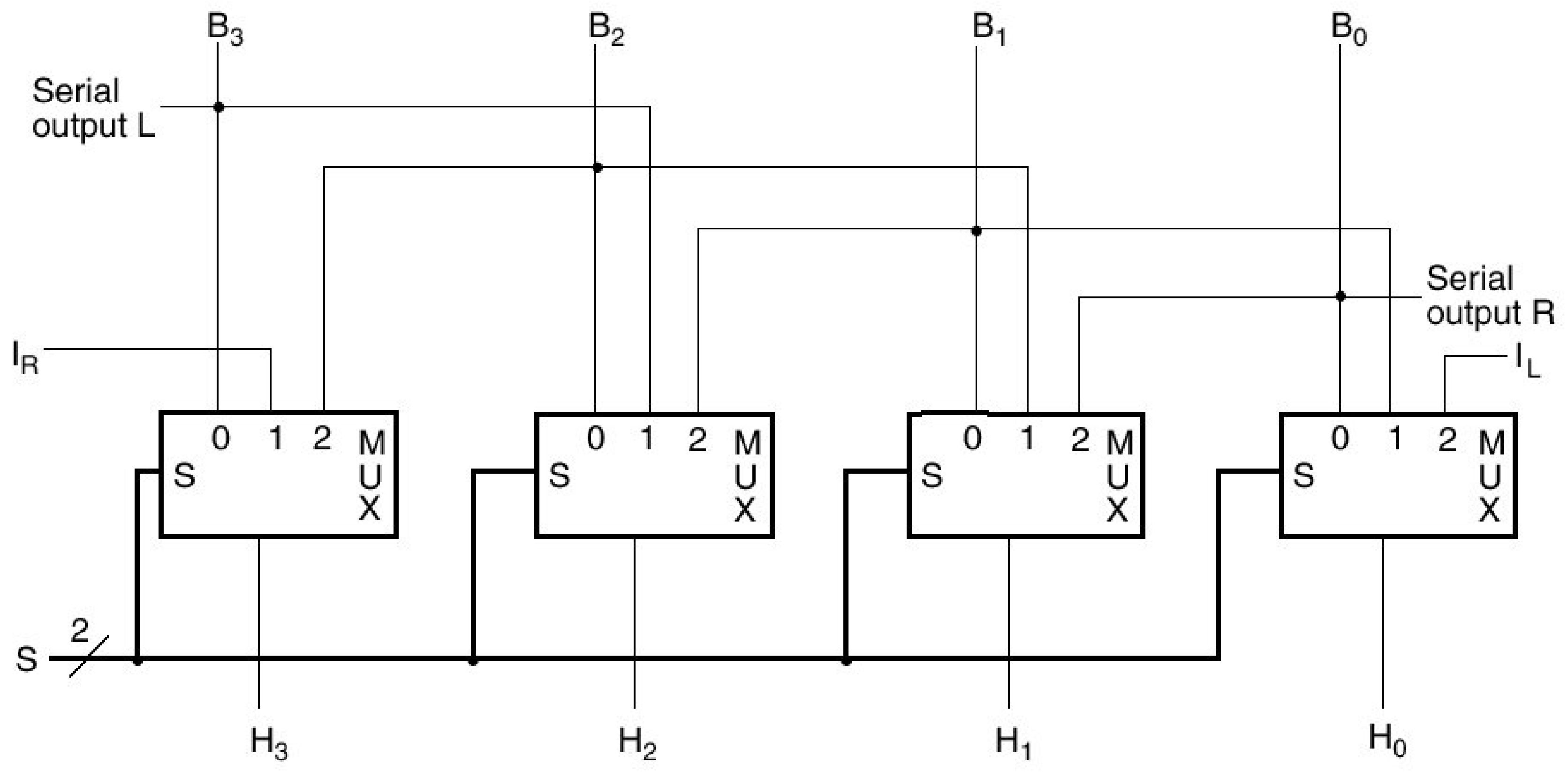
Adder

- This adder this gives us a fast combinational ALU with the following functionality:

Select				Output	
S_2	S_1	S_0	C_{in}		
0	0	0	0	G = A	TRANSFER
0	0	0	1	G = A + 1	INCREMENT
0	0	1	0	G = A + B	ADD
0	0	1	1	G = A + B +1	ADD WITH C
0	1	0	0	G = A + $\overline{B}$	A plus 1’s C.B
0	1	0	1	G = A + B +1	SUBTRACT
0	1	1	0	G = A - 1	DECREMENT
0	1	1	1	G = A	TRANSFER
1	0	0	X	G = A $\wedge$ B	AND
1	0	1	X	G = A $\vee$ B	OR
1	1	0	X	G = A $\oplus$ B	XOR
1	1	1	X	G = A	NOT

4-Bit SR/SL Shifter Unit

- For speed of execution the shifter unit is always implemented as a combinational circuit based on a MUX:



B_3
B_2
B_1
B_0
Serial output L
Serial output R
I_R
I_L
0 1 2 MUX S
0 1 2 MUX S
0 1 2 MUX S
0 1 2 MUX S
S
2
H_3
H_2
H_1
H_0