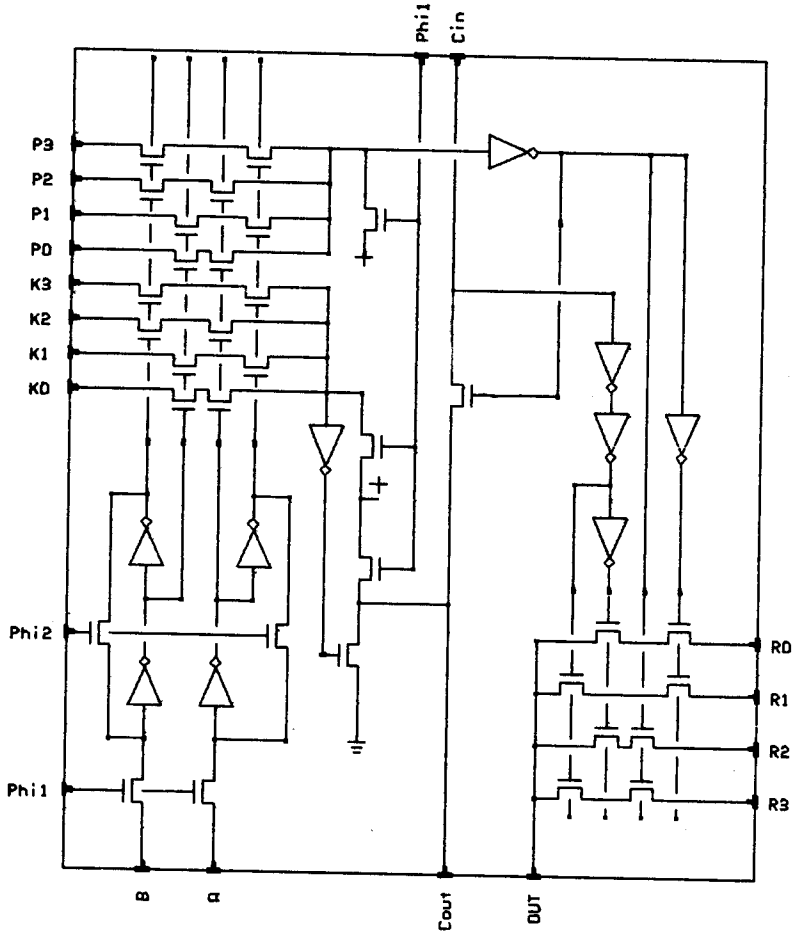
$\overline{A \& B}$	0	8	12	0	Bit-wise NAND
Zero	0	0	0	0	Zero function
One	0	0	4	0	One function
$\overline{A=>B}$	0	2	1	0	$A\overline{B}$
$\overline{B=>A}$	0	4	1	0	$\overline{A}B$
A	0	10	3	0	Just A
B	0	12	3	0	Just B

The numbers are decimal representations of the four-bit binary numbers that are fed to the function blocks, where P0, K0, and R0 are the least significant bits. All the data words are assumed to be in two's complement form. Logical operations allow 0 or 1 for LSB carry-in; however, a 0 input ensures a 0 output in carry out. Other operations such as shifts and mixed logic functions are possible.

TESSELATION

Abuts horizontally, without overlap.

LOGIC DIAGRAM



NAME

ALU—ALU bit-slice

SYNOPSIS

ALU is one bit of the OM2 Data Path Chip ALU.

PROPERTIES

Size

45 x 254 λ

Interface

A	input,	switched	3 □ C_g
B	input,	switched	3 □ C_g

K0 is typical of the function block lines:

K0	input,	switched	4.25 □ C_g
CIN	input,	switched	3.75 □ C_g
COUT	output,	precharged	4.5 □ C_g
OUT	output,	switch-logic	4.5 □ C_g

DC Current

0.4 mA.

DESCRIPTION

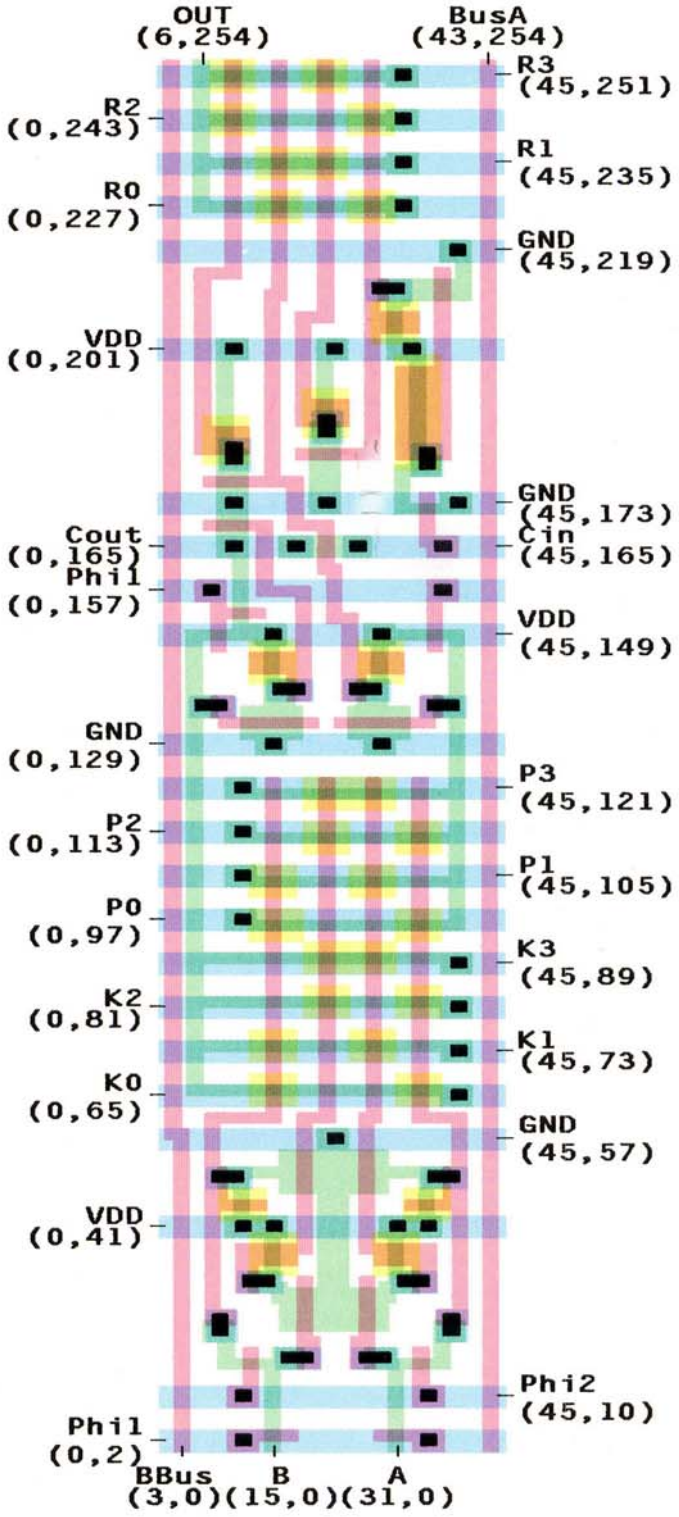
This is one bit of the OM2 Data Path ALU. It employs a very general structure to generate arbitrary functions of its two inputs and a carry-in. The carry is propagated using a Manchester-type carry chain. Extensive use is made of precharging.

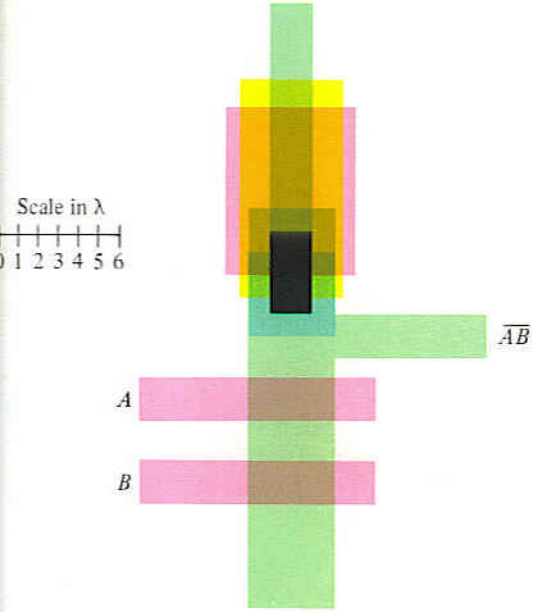
Note that the output comes directly out of the R function block. It should be buffered before it is used anywhere.

ALU Operations

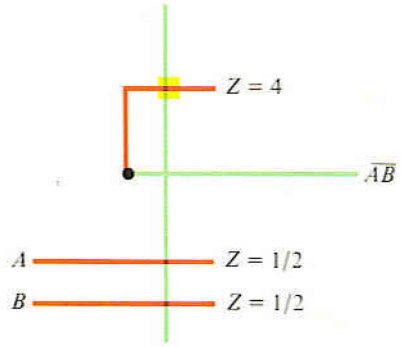
Here is a table of ALU operations that have been checked by simulation:

OP	K	P	R	LSB Cin	Operation
A+B	14	9	6	—	Add with carry
A−B	11	6	6	1	$A+\overline{B}+1$
B−A	13	6	6	1	$\overline{A}+B+1$
−A	5	10	6	1	$\overline{A}+1$
−B	3	12	6	1	$\overline{B}+1$
A+Cin	10	5	6	—	Increment A if LSB Cin=1
B+Cin	12	3	6	—	Increment B if LSB Cin=1
A-Cin	5	10	9	—	Decrement A if LSB Cin=1
B-Cin	3	12	9	—	Decrement B if LSB Cin=1
A *and* B	0	8	3	0	Bit-wise AND
A *or* B	0	1	4	0	Bit-wise OR
A *xor* B	0	6	3	0	Bit-wise XOR
$\overline{A}$	0	5	3	0	Not A
$\overline{B}$	0	3	3	0	Not B
$\overline{A \textit{ or } B}$	0	1	1	0	Bit-wise NOR
$\overline{A \textit{ xor } B}$	0	9	1	0	XNOR
B=>A	0	4	12	0	B implies A
A=>B	0	2	12	0	A implies B

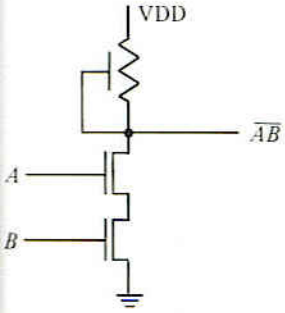

OUT
(6,254)
BusA
(43,254)
R3
(45,251)
R2
(0,243)
R1
(45,235)
R0
(0,227)
GND
(45,219)
VDD
(0,201)
GND
(45,173)
Cout
(0,165)
Cin
(45,165)
Phi1
(0,157)
VDD
(45,149)
GND
(0,129)
P3
(45,121)
P2
(0,113)
P1
(45,105)
P0
(0,97)
K3
(45,89)
K2
(0,81)
K1
(45,73)
K0
(0,65)
GND
(45,57)
VDD
(0,41)
Phi2
(45,10)
Phi1
(0,2)
BBus
(3,0)
B
(15,0)
A
(31,0)



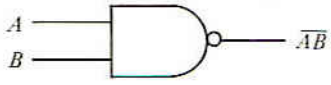
(a) NAND gate layout geometry.



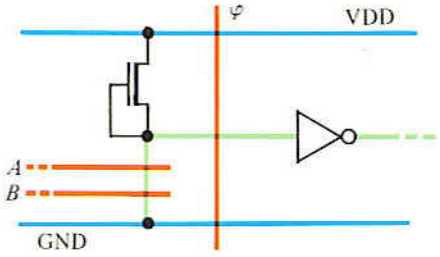
(b) NAND gate topology (stick diagram).



(c) NAND gate circuit diagram.

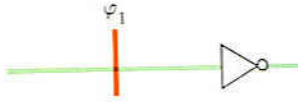


(d) NAND gate logic symbol.

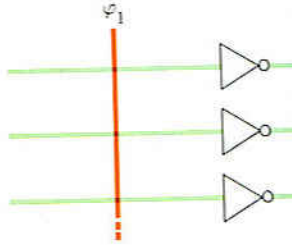


(e) Example of mixed notation.

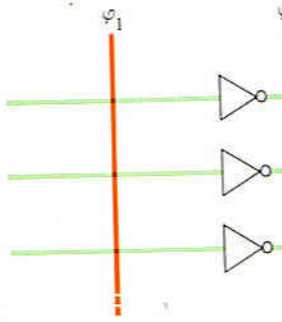
PLATE 4 Notation



(a) Shift reg



(b) A



(c)

PLAT