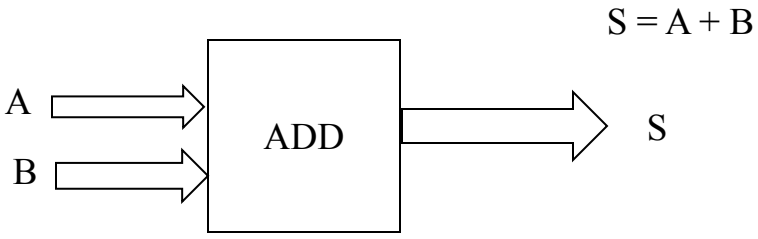
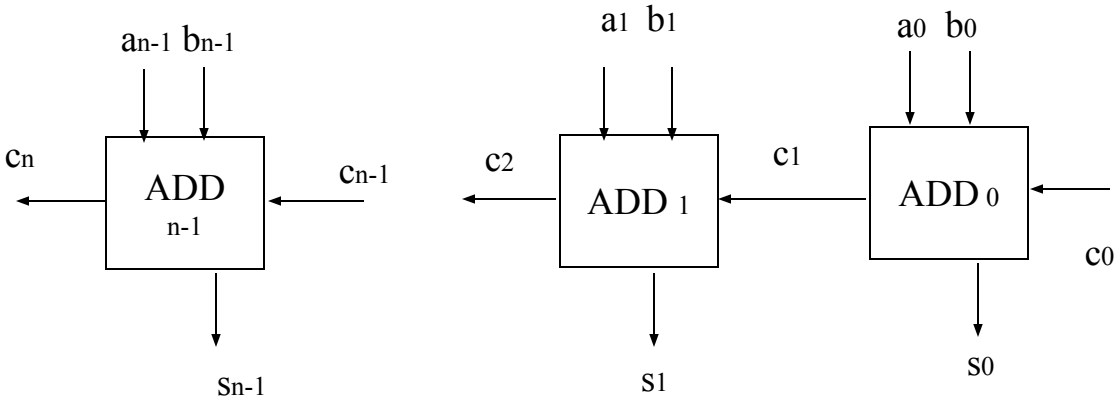
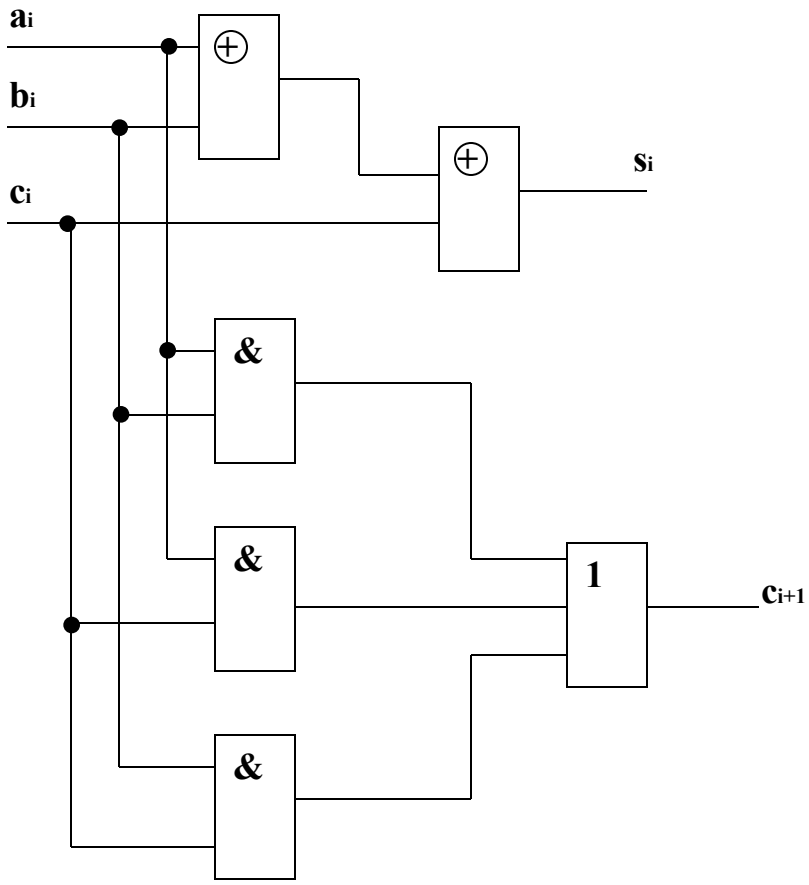
Summaator *Adder*



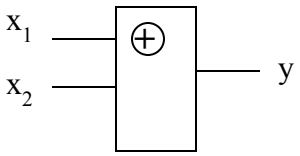


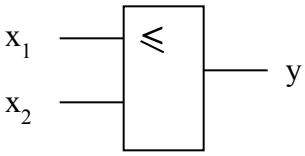
a_i	b_i	c_i	s_i	c_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

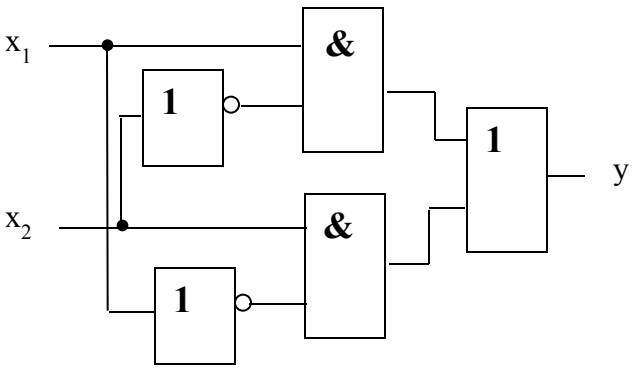


x1	x2	y
0	0	0
0	1	1
1	0	1
1	1	0

$$y = x_1 \overline{x_2} + \overline{x_1} x_2$$





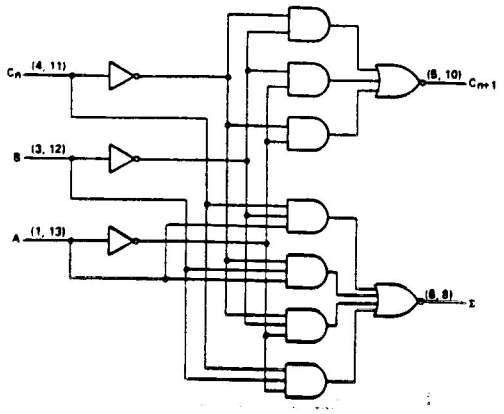


TTL
MSI

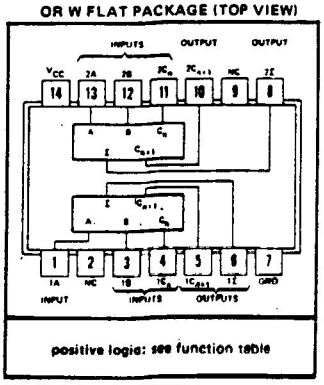
TYPES SN54H183, SN74H183
DUAL CARRY-SAVE FULL ADDERS

- For Use in High-Speed Wallace-Tree Summing Networks
- Sum and Carry Propagation Delays Both Average Typically 11 ns
- High-Speed, High-Fan-Out Darlington Outputs
- Input Clamping Diodes Simplify System Design
- Compatible with Most TTL and DTL Circuits
- Typical Power Dissipation . . . 110 mW per Bit

functional block diagram (each adder)



J OR N DUAL-IN-LINE
OR W FLAT PACKAGE (TOP VIEW)



NC—No internal connection

FUNCTION TABLE
(EACH ADDER)

INPUTS			OUTPUTS	
C_n	B	A	Σ	C_{n+1}
L	L	L	L	L
L	L	H	H	L
L	H	L	H	L
L	H	H	L	H
H	L	L	H	L
H	L	H	L	H
H	H	L	L	H
H	H	H	H	H

H = high level, L = low level

TYPES SN5482, SN7482
2-BIT BINARY FULL ADDERS

For applications in:

- Digital Computer Systems
- Data-Handling Systems
- Control Systems

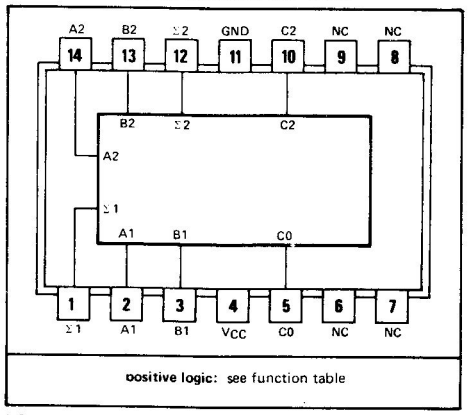
logic

FUNCTION TABLE

INPUTS				OUTPUTS					
				WHEN C0 = L			WHEN C0 = H		
A1	B1	A2	B2	Σ1	Σ2	C2	Σ1	Σ2	C2
L	L	L	L	L	L	L	H	L	L
H	L	L	L	H	L	L	L	H	L
L	H	L	L	H	L	L	L	H	L
H	H	L	L	L	H	L	H	H	L
L	L	H	L	L	H	L	H	H	L
H	L	H	L	H	H	L	L	L	H
L	H	H	L	H	H	L	L	L	H
H	H	H	L	L	L	H	H	L	H
L	L	L	H	L	H	L	H	H	L
H	L	L	H	H	H	L	L	L	H
L	H	L	H	H	H	L	L	L	H
H	H	L	H	L	L	H	H	L	H
L	L	H	H	L	L	H	H	L	H
H	L	H	H	H	L	H	L	H	H
L	H	H	H	H	L	H	L	H	H
H	H	H	H	L	H	H	H	H	H

H = high level, L = low level

J OR N DUAL-IN-LINE OR
W FLAT PACKAGE (TOP VIEW)

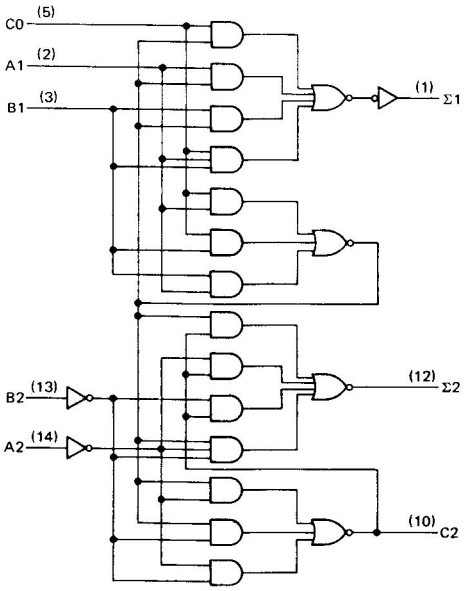


NC—No internal connection

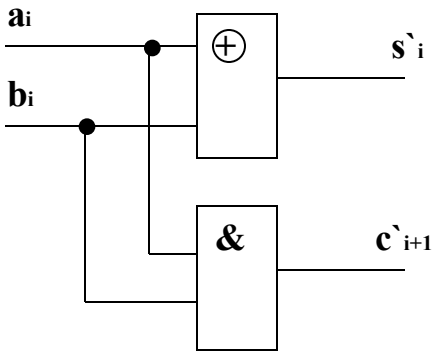
description

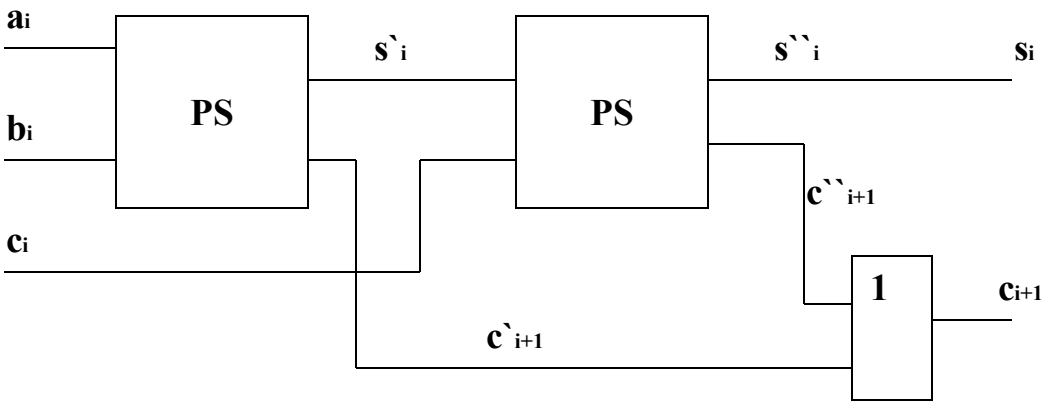
These full adders perform the addition of two 2-bit binary numbers. The sum (Σ) outputs are provided for each bit and the resultant carry (C2) is obtained from the second bit. Designed for medium-to-high-speed, multiple-bit, parallel-add/serial-carry applications, these circuits utilize high-speed, high-fan-out transistor-transistor logic (TTL) and are compatible with both DTL and TTL logic families. The implementation of a single-inversion, high-speed, Darlington-connected serial-carry circuit within each bit minimizes the necessity for extensive "look-ahead" and carry-cascading circuits.

functional block diagram

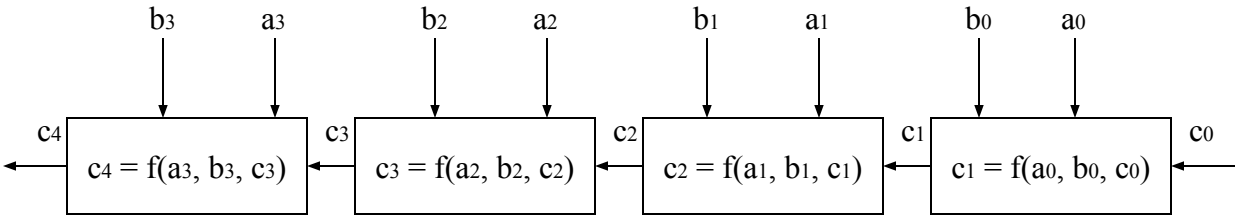


a_i	b_i	$s_i`$	$c_{i+1}`$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

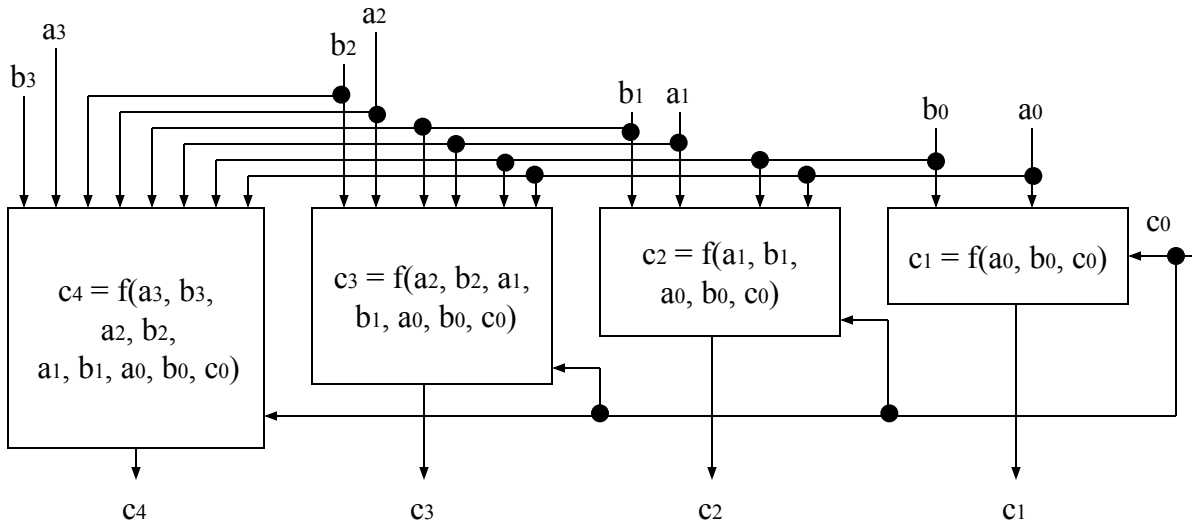


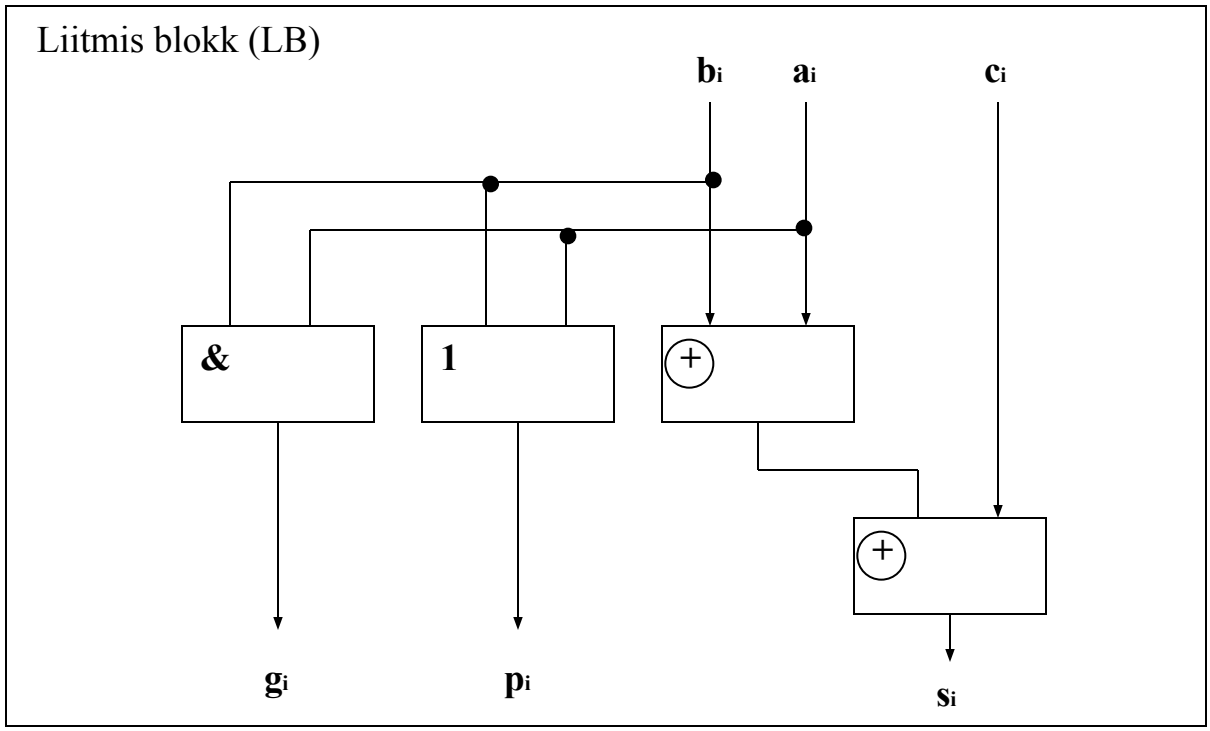


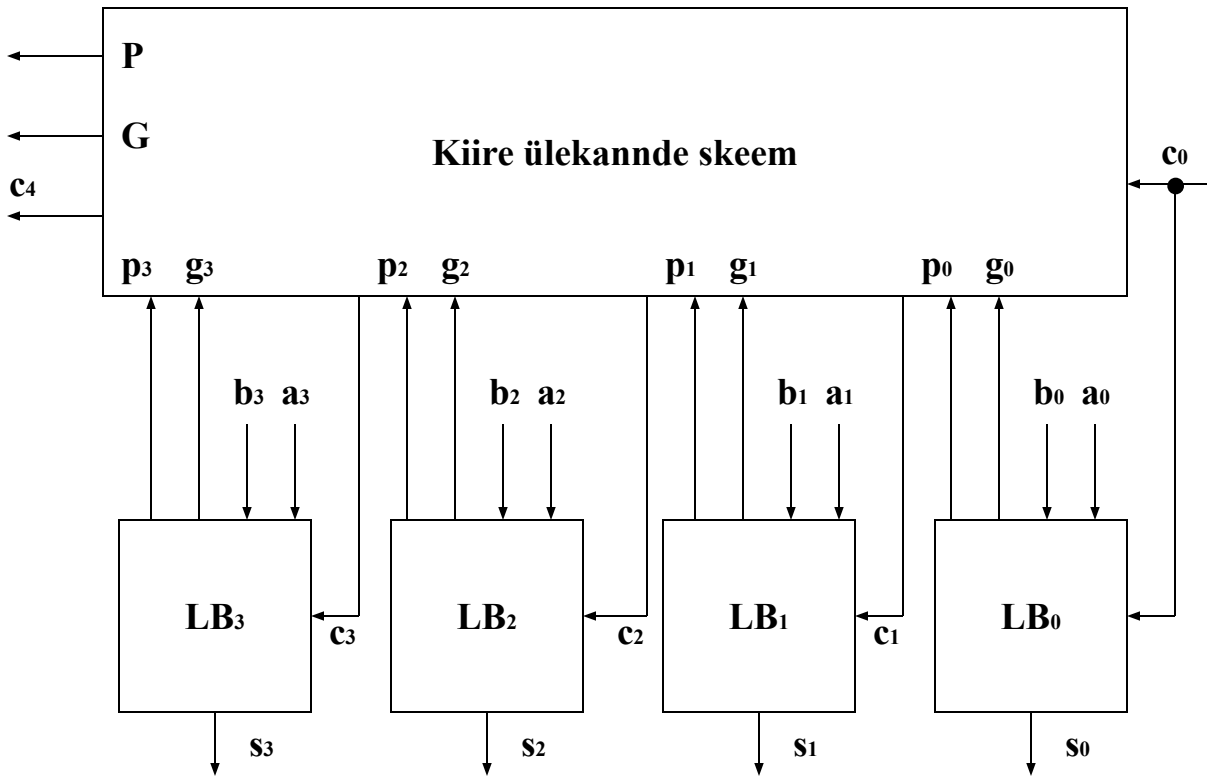
Järjestikülekanne.



Paralleelülekanne.



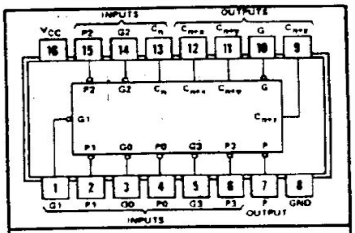

Liitmis blokk (LB)
bi
ai
ci
&
1
+
+
gi
pi
si


P
G
c4
Kiire ülekannde skeem
c0
p3 g3 p2 g2 p1 g1 p0 g0
b3 a3
b2 a2
b1 a1
b0 a0
LB3
LB2
LB1
LB0
c3
c2
c1
s3
s2
s1
s0

PIN DESIGNATIONS

DESIGNATION	PIN NOS.	FUNCTION
G0, G1, G2, G3	3, 1, 14, 5	ACTIVE-LOW CARRY GENERATE INPUTS
P0, P1, P2, P3	4, 2, 15, 6	ACTIVE-LOW CARRY PROPAGATE INPUTS
C_n	13	CARRY INPUT
C_{n+x}, C_{n+y}, C_{n+z}	12, 11, 9	CARRY OUTPUTS
G	10	ACTIVE-LOW CARRY GENERATE OUTPUT
P	7	ACTIVE-LOW CARRY PROPAGATE OUTPUT
V_{CC}	16	SUPPLY VOLTAGE
GND	8	GROUND

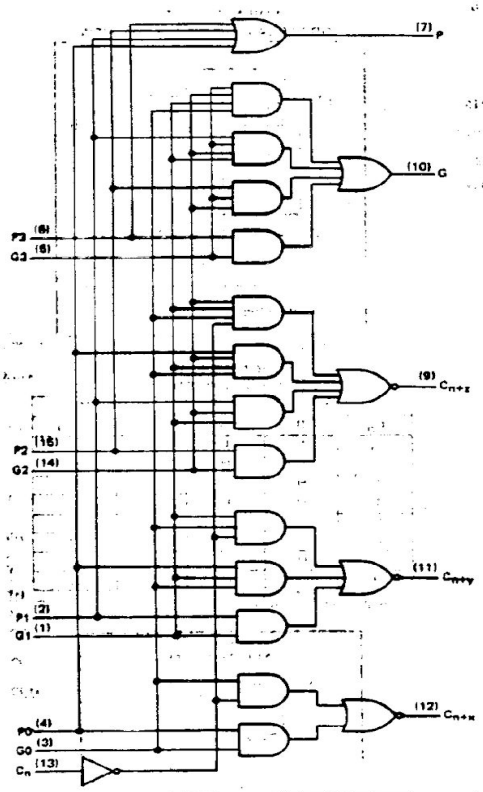
J OR N DUAL-IN-LINE
OR W FLAT PACKAGE (TOP VIEW)



logic: see description

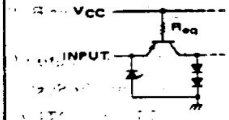
TYPE	TYPICAL AVERAGE PROPAGATION DELAY TIME	TYPICAL POWER DISSIPATION
'182	13 ns	180 mW
'S182	7 ns	260 mW

functional block diagram and schematics of inputs and outputs



'182

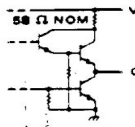
EQUIVALENT OF EACH INPUT



INPUT	Req NOM
C_n	2.8 kΩ
P3	1.4 kΩ
P2	940 Ω
P0, P1, G3	700 Ω
G0, G4	400 Ω
G1	350 Ω

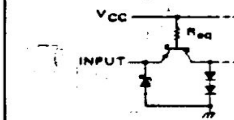
'182

TYPICAL OF ALL OUTPUTS



'S182

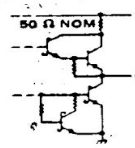
EQUIVALENT OF EACH INPUT



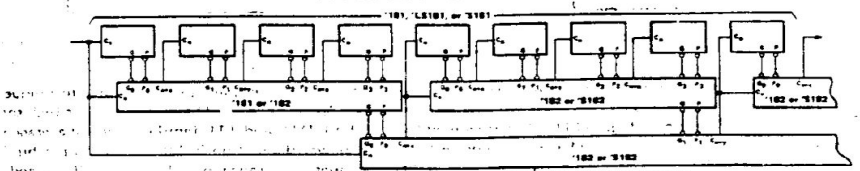
INPUT	Req NOM
C_n	2.8 kΩ
P3	1.4 kΩ
P2	940 Ω
P0, P1, G3	700 Ω
G0, G4	400 Ω
G1	350 Ω

'S182

TYPICAL OF ALL OUTPUTS



TYPICAL APPLICATION DATA

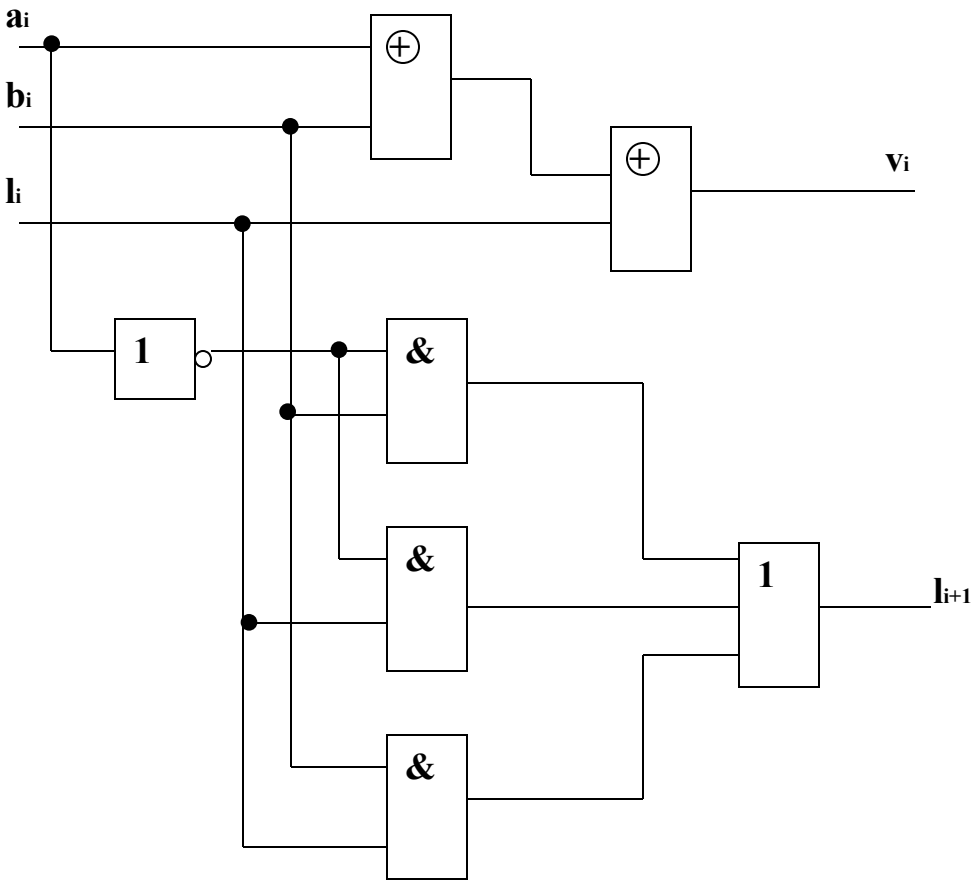


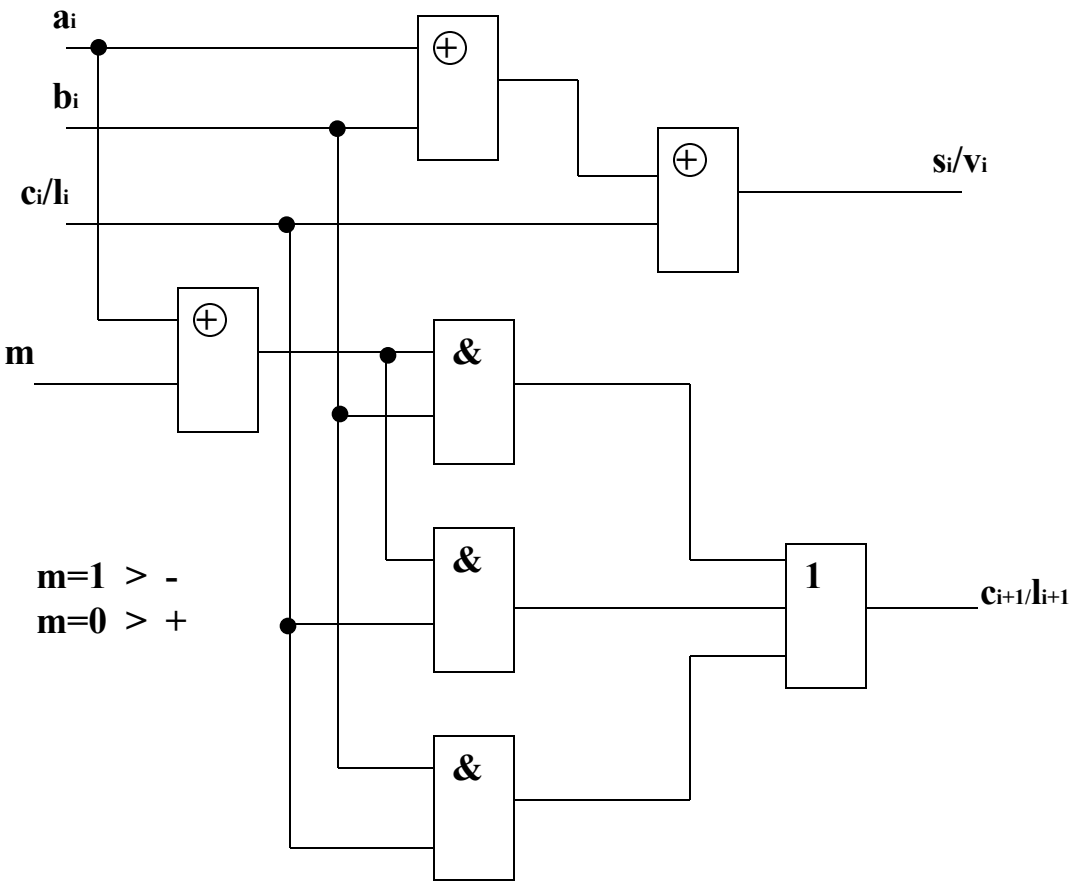
64-BIT ALU, FULL-CARRY LOOK-AHEAD IN THREE LEVELS

A and B inputs and F outputs of '181, 'LS181, and 'S181 are not shown.

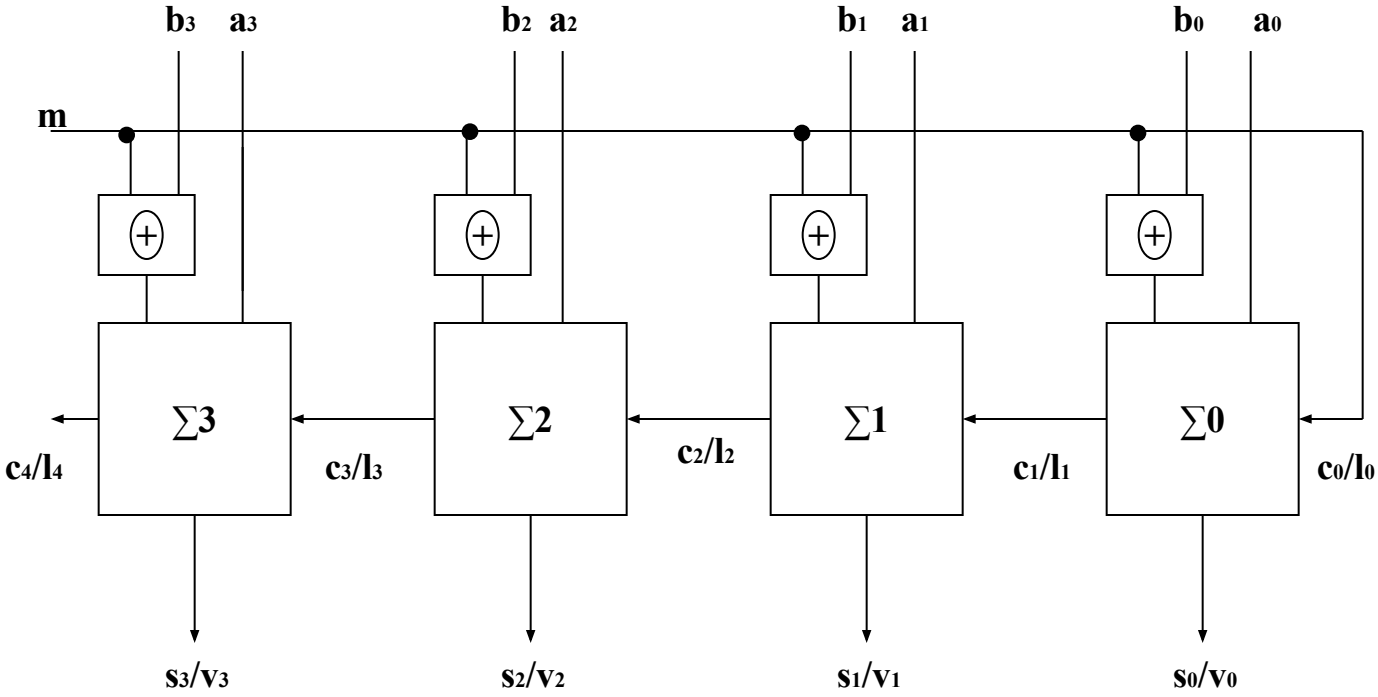
Lahutaja *Subtractor*

a_i	b_i	l_i	v_i	l_{i+1}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1




ai
bi
ci/li
m
m=1 > -
m=0 > +
si/vi
ci+1/li+1
&
&
&
1

m=1 > -
m=0 > +


b3 a3
b2 a2
b1 a1
b0 a0
m
∑3
∑2
∑1
∑0
c4/l4
c3/l3
c2/l2
c1/l1
c0/l0
s3/v3
s2/v2
s1/v1
s0/v0