Microarchitecture Specification

CSE 529 - 1988

MAC Chip

Rani Khatwani
Dahmane Dahmani

Technical Report CS/E-86-037
October 1, 1988

Oregon Graduate Center
19600 S.W. von Neumann Drive
Beaverton, Oregon 97006-1999

Microarchitecture Specification
CSE 529 - 1988

MAC Chip

Rani Khatwani, Dahmane Dahmani

October 1, 1988

1. Introduction

The purpose of this document is to describe the Microarchitectural Specifications of the 16x8 Bit Multiplier/Accumulator Chip (MAC). This project is a part of the course, CSE 529, offered at the Oregon Graduate Center. The design will be used as a module on a larger chip which is a part of the research project on the study of cognitive behaviour being conducted at the Oregon Graduate Center.

The MAC Chip comprises of a multiplier and an accumulator. Multiplication is often an essential function in digital systems. Multipliers can be designed in three different ways, namely: serial, serial-parallel and parallel (more commonly known as array-multipliers). The tradeoff between these three designs is based upon factors such as speed, throughput and area of the chip. Since our target was design of high speed MAC, we choose to use parallel multiplier and a fast adder for the accumulator.

2. General Description

2.1. Multiplier

The MAC chip multiplies two sign bit numbers X and Y. The multiplicand X has 16 bits (15 bits magnitude and 1 sign bit), while the multiplier Y has 8 bits (7 bits magnitude and 1 sign bit). The product obtained from the multiplication will be a 23 bit word with the MSB representing the sign bit. The product is added to the previous sum and stored in the accumulator. This result is also available at the output register.

The multiplication is based on the following algorithm:

Let X and Y be two signed binary integers:

$$X = X_s X_{n-1} \ldots\ldots X_0 \quad \textit{Multiplicand}$$

$$Y = Y_s Y_{m-1} \ldots\ldots Y_0 \quad \textit{Multiplier}$$

where X_s and Y_s are the sign bits of the multiplicand and the multilipier respectively. The product is formed as follows:

a) The sign of the product is determined from the signs of the multiplicand and the multiplier. If they are alike, the sign of the product is plus. If they are unlike, the sign of the product is minus, i.e.,

$$P_s = X_s + Y_s$$

b) The magnitude of the product is given by:

$$P_r = (\sum_{i=0}^{n-1} X_i 2^i)(\sum_{j=0}^{m-1} Y_j 2^j)$$

$$= (\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}(X_i Y_j)\, 2^{i+j})$$

$$= (\sum_{k=0}^{m+n-1} P_k 2^k)$$

where P_k's are the partial product terms. There are mn partial product terms which are produced in parallel. So in our design there will be 15x7 partial product terms and the product will be a 23 bit word with the MSB representing the sign bit.

The functionality of the parallel multiplier can be explained by the following example :

$$\begin{array}{cccc} X_3 & X_2 & X_1 & X_0 \\ & Y_2 & Y_1 & Y_0 \end{array}$$

$$\begin{array}{cccc} X_3Y_0 & X_2Y_0 & X_1Y_0 & X_0Y_0 \\ X_3Y_1 & X_2Y_1 & X_1Y_1 & X_0Y_1 \\ X_3Y_2 & X_2Y_2 & X_1Y_2 & X_0Y_2 \end{array}$$

$$P_6 \quad P_5 \quad P_4 \quad P_3 \quad P_2 \quad P_1 \quad P_0$$

2.2. MAC Equation

The MAC equation is given by :

RSLT = RSLT + P

where, RSLT is the result register and P is the signed product of X and Y.

2.3. MAC Algorithm

The algorithm of the MAC chip is given by :

Reset

Do (forever)

{

Load X, Y

Start multiplication

Start 2's complement of the product

Add to previous result and store in accumulator

Take 2's complement of accumulator

Output result

}

3. External Interface

3.1. External Input/Output Specification

Pin Specifications for the chip are as follows :

a) Input: 24 pins (16 pins for the Multiplicand and 8 pins for the multiplier).
b) Output: 23 pins (includes 1 pin for signout).
c) Load: Loads the input registers X and Y when high.
d) Overflow: 1 pin (This goes high when there is an overflow and the MAC gets reset to zero).
e) Reset: 1 pin (To reset the whole chip)
f) Clock: 2 pins (phase1 and phase2)
g) V_{cc}: 3 pins
h) GND: 3 pins

3.2. System Architecture

The MAC is divided into four modules, namely, the input register module (inmod_C), the array multiplier module (mulmod_C), the add/acc module (accmod_C), the output module (outmod_C).

3.3. MAC External Signals

This section describes all the external signals of the MAC.

3.3.1. Bidirectional I/O

eX_B : Input - 16 bit data bus representing the multiplicand. The MSB is the sign bit.

eY_B : Input - 8 bit data bus representing the multiplier. The MSB is the sign bit.

rslt : Output - 22 bit data bus representing the result of the MAC.

signout : Sign bit of the MAC result.

ovrflwout : Output - high in case of an overflow.

3.3.2. Clocks

clk_1 : Input - Phase 1 of a two phase non-overlapping clock.

clk_2 : Input - Phase 2 of a two phase non-overlapping clock.

3.3.3. Reset and Load

load : Input - 1 bit, when asserted the inputs are loaded in the input module.

reset : Input - 1 bit, resets the MAC chip

3.3.4. Power & GND

Vcc = 5 volt pins

GND = ground pins

4. Internal Architecture

The chip is divided into four modules which together perform the function of the MAC. Fig 1 shows a block diagram of the MAC with its modules.

4.1. Input Register Module (inmod_C)

This module consists of 2 registers, namely, register X with 16 bit bidirectional inputs and register Y with 8 bit bidirectional inputs.

4.1.1. Signals to/from Module off Chip

ph1 : Input, phase 1 of clock signal.

ph2 : Input, phase 2 of clock signal.

reset : Input resets the inmod_C.

eX_B : Input (16 bits), multiplicand.

eY_B : Input (8 bits), multiplier.

load : Input - if high, loads the registers X and Y with eX_B and eY_B respectively in ph1.

4.1.2. Signals to/from Module on Chip

ph2I : Input, phase 2 inverted.

X_B2 : Output (16 bits) to mulmod_C (the MSB is sign bit).

Y_B2 : Output (8 bits) to mulmod_C (the MSB is sign bit).

ovrflw_1 : Input, 1 bit, overflow signal from the accmod_C (resets the inmod_C to zero).

4.2. Array Multiplier Module (mulmod_C)

This module calculates the final product (fp_B2) and sign (msign_2) of X_B2 and Y_B2. It uses the Wallace tree architecture. The two MSB from the inmod_C are fed into an XOR gate to give the sign bit, msign2, of the final product.

The multiplier module has four Wallace stages consisting of full adders (FA) and half adders (HA) and a Manchester Carry Lookahead Adder in the last stage. The table below shows the number of full and half adders used in the multiplier module.

Stage	Number of FA	Number of HA
0	27	4
1	14	1
2	13	0
3	13	2
Total	67	7

TABLE 1: FA and HA in the mulmod_C.

The last stage is a Manchester Carry Lookahead Adder and its operation is explained in section 3.4.3.

4.2.1. Signals to/from Module off Chip

ph1 : Input, phase 1 used to evaluate the Manchester adder.

4.2.2. Signals to/from Module on Chip

ph2I : Input, Inverted phase 2 used to "precharge" the Manchester adder circuit.

X_B2 : Input (16 bits), multiplicand, from inmod_C.

Y_B2 : Input (8 bits), multiplier, from inmod_C.

fp_B2 : Output (22 bits), final product of multiplier to tcamod_C.

msign_2 : Output (1 bit), sign bit, to tcamod_C.

4.2.3. Operation of the mulmod_C

The multiplier uses the Wallace tree architecture and a Manchester Carry Lookahead Adder in the last stage. A block diagram of the Multiplier and the Manchester Adder are shown in Fig 2.

4.3. Adder/Accumulator Module (accmod_C)

This module converts the final product output of the multiplier into the 2's complement format and adds it to the number stored in the accumulator (namely, register B).

4.3.1. Signals to/from Module off Chip

ph1 : Input, phase 1 clock.

ph1I : Input, inverted phase 1 clock.

ph2 : Input, phase 2 clock.

ph2I : Input, inverted phase 2 clock.

reset : Resets the accmod_C to zero.

4.3.2. Signals to/from Module on Chip

fp_B2 : Input (22 bits), final product from mulmod_C.

msign_2 : Input (1 bit), enables the one's complement circuitry.

ovrflw : Output (1 bit), high in case of an overflow.

sum : Output (23 bits), to outmod_C. The MSB is the sign bit.

4.3.3. Operation of accmod_C

The accmod_C module consists of various submodules, namely, the onecomp_C, the Areg23_C, Breg23_C, the Manadder_C and an sreg23_C.

The value of msign_2 serves as the control logic for the onecomp_C. If it is 1, it means that the number fp_B2 is negative and the 1's complement circuitry gets enabled. If it is zero it indicates that fp_B2 is positive and its 1's complement is the number itself. The output of the one's complement cell

(product) then gets added to the previous sum stored in the accumulator. We use a Carry Lookahead Manchester carry chain to implement the adder. A block diagram of the adder is shown in Fig 3.

The adder will add two numbers, the final product of the multiplier and the contents of the accumulator (i.e the previous sum). The numbers are 23 bit long, with the most significant bit being the sign bit. The two numbers are in the 2's complement format and the result of the adder will also be in the 2's complement format.

The carry C22 (generated out of the sum of the 22nd bits) and carry C23 (generated out of the sum of the 23rd bits) are of special interest. They specify if there is an overflow in the result (i.e result is incorrect).

The block diagram of the accmod_C is shown in Fig 4. with some examples of addition given in the appendix.

4.4. Output Module (outmod_C)

This module consists of the one's complement cell, the incr22_C and a dregB_C.

4.4.1. Signals to/from Module off Chip

ph1 : Input, phase 1 clock.

ph1I : Input, phase 1 inverted.

ph2 : Input, phase 2 clock.

ph2I : Input, phase 2 inverted.

rslt : Output (22 bits), the final product of the two input numbers.

signout: Output (1 bit), the sign of the final product.

ovrflwout : Output (1 bit), goes high in case of an overflow to indicate that the output result is incorrect.

reset : Input (1 bit), resets the dregB_C to zero.

4.4.2. Signals to/from Module on Chip

sum : Input (23 bits), from accmod_C (the MSB is the sign bit).

4.4.3. Operation of outmod_C

A block diagram of the outmod_C is shown in Fig 5. The onecomp_C takes the one's complement of the product from the accmod_C. The incr22_C increments the number by one, depending on whether the signin bit is zero or one. This final product in the sign-magnitude format is then stored inside the dregB_C which outputs the number in ph1.

5. Timing/Speed and Area

The target speed of the chip is 12 Mhz. The whole process of multilication right from the time the INMOD registers are loaded to the time the result is available at the output takes 3.5 cycles. The timing diagram of the operation of the MAC is shown in Fig 6.

The frame used for this chip was a 64 pin, 6900x6800 frame (64p69x68). The pin numbers with their corresponding names is given in Fig 7.

APPENDIX

SOME EXAMPLES :

Assume we are adding two numbers, each 5 bits long, with the MSB being the sign bit. All negative numbers are in 2's complement format.

Case 1:

```
  0 1 1 1 1        15
+ 0 1 1 0 1      + 13
0 1 1 1 0 0        28
```

C23= 0 C22=1 ==========> OVERFLOW (incorrect result)

Case 2:

```
  0 0 0 1 1        3
+ 0 1 0 0 1      + 9
0 0 1 1 0 0        12
```

C23=0 C22=0 ========> Result correct (01100= + 12)

Case 3:

```
  1 0 0 0 1       - 15
+ 0 0 1 1 0     +   6
0 1 0 1 1 1       - 9
```

C23=0 C22=0 ==========> Result correct (10111= -9)

Case 4:

```
  1 0 1 1 1       - 9
+ 1 1 1 0 0     + - 4
1 1 0 0 1 1       - 13
```

C23= 1 C22=1 ==========> Result correct (throw C23 and 10011= - 13)

Microarchitecture Specification
CSE 529 - 1988

MAC Chip

Rani Khatwani, Dahmane Dahmani

October 1, 1988

1. Introduction

The purpose of this document is to describe the Microarchitectural Specifications of the 16x8 Bit Multiplier/Accumulator Chip (MAC). This project is a part of the course, CSE 529, offered at the Oregon Graduate Center. The design will be used as a module on a larger chip which is a part of the research project on the study of cognitive behaviour being conducted at the Oregon Graduate Center.

The MAC Chip comprises of a multiplier and an accumulator. Multiplication is often an essential function in digital systems. Multipliers can be designed in three different ways, namely: serial, serial-parallel and parallel (more commonly known as array-multipliers). The tradeoff between these three designs is based upon factors such as speed, throughput and area of the chip. Since our target was design of high speed MAC, we choose to use parallel multiplier and a fast adder for the accumulator.

2. General Description

2.1. Multiplier

The MAC chip multiplies two sign bit numbers X and Y. The multiplicand X has 16 bits (15 bits magnitude and 1 sign bit), while the multiplier Y has 8 bits (7 bits magnitude and 1 sign bit). The product obtained from the multiplication will be a 23 bit word with the MSB representing the sign bit. The product is added to the previous sum and stored in the accumulator. This result is also available at the output register.

The multiplication is based on the following algorithm:

Let X and Y be two signed binary integers:

$$X = X_s X_{n-1} \ldots\ldots X_0 \quad \textit{Multiplicand}$$

$$Y = Y_s Y_{m-1} \ldots\ldots Y_0 \quad \textit{Multiplier}$$

where X_s and Y_s are the sign bits of the multiplicand and the multilipier respectively. The product is formed as follows:

a) The sign of the product is determined from the signs of the multiplicand and the multiplier. If they are alike, the sign of the product is plus. If they are unlike, the sign of the product is minus, i.e.,

$$P_s = X_s + Y_s$$

b) The magnitude of the product is given by:

$$P_r = (\sum_{i=0}^{n-1} X_i 2^i)(\sum_{j=0}^{m-1} Y_j 2^j)$$

$$= (\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}(X_i Y_j)\, 2^{i+j})$$

$$= (\sum_{k=0}^{m+n-1} P_k 2^k)$$

where P_k's are the partial product terms. There are mn partial product terms which are produced in parallel. So in our design there will be 15x7 partial product terms and the product will be a 23 bit word with the MSB representing the sign bit.

The functionality of the parallel multiplier can be explained by the following example :

$$\begin{array}{cccc} X_3 & X_2 & X_1 & X_0 \\ & Y_2 & Y_1 & Y_0 \end{array}$$

$$\begin{array}{cccccc} & & X_3Y_0 & X_2Y_0 & X_1Y_0 & X_0Y_0 \\ & X_3Y_1 & X_2Y_1 & X_1Y_1 & X_0Y_1 & \\ X_3Y_2 & X_2Y_2 & X_1Y_2 & X_0Y_2 & & \end{array}$$

$$P_6 \quad P_5 \quad P_4 \quad P_3 \quad P_2 \quad P_1 \quad P_0$$

2.2. MAC Equation

The MAC equation is given by :

RSLT = RSLT + P

where, RSLT is the result register and P is the signed product of X and Y.

2.3. MAC Algorithm

The algorithm of the MAC chip is given by :

Reset

Do (forever)

{

Load X, Y

Start multiplication

Start 2's complement of the product

Add to previous result and store in accumulator

Take 2's complement of accumulator

Output result

}

3. External Interface

3.1. External Input/Output Specification

Pin Specifications for the chip are as follows :

a) Input: 24 pins (16 pins for the Multiplicand and 8 pins for the multiplier).
b) Output: 23 pins (includes 1 pin for signout).
c) Load: Loads the input registers X and Y when high.
d) Overflow: 1 pin (This goes high when there is an overflow and the MAC gets reset to zero).
e) Reset: 1 pin (To reset the whole chip)
f) Clock: 2 pins (phase1 and phase2)
g) V_{cc}: 3 pins
h) GND: 3 pins

3.2. System Architecture

The MAC is divided into four modules, namely, the input register module (inmod_C), the array multiplier module (mulmod_C), the add/acc module (accmod_C), the output module (outmod_C).

3.3. MAC External Signals

This section describes all the external signals of the MAC.

3.3.1. Bidirectional I/O

eX_B : Input - 16 bit data bus representing the multiplicand. The MSB is the sign bit.

eY_B : Input - 8 bit data bus representing the multiplier. The MSB is the sign bit.

rslt : Output - 22 bit data bus representing the result of the MAC.

signout : Sign bit of the MAC result.

ovrflwout : Output - high in case of an overflow.

3.3.2. Clocks

clk_1 : Input - Phase 1 of a two phase non-overlapping clock.

clk_2 : Input - Phase 2 of a two phase non-overlapping clock.

3.3.3. Reset and Load

load : Input - 1 bit, when asserted the inputs are loaded in the input module.

reset : Input - 1 bit, resets the MAC chip

3.3.4. Power & GND

Vcc = 5 volt pins

GND = ground pins

4. Internal Architecture

The chip is divided into four modules which together perform the function of the MAC. Fig 1 shows a block diagram of the MAC with its modules.

4.1. Input Register Module (inmod_C)

This module consists of 2 registers, namely, register X with 16 bit bidirectional inputs and register Y with 8 bit bidirectional inputs.

4.1.1. Signals to/from Module off Chip

ph1 : Input, phase 1 of clock signal.

ph2 : Input, phase 2 of clock signal.

reset : Input resets the inmod_C.

eX_B : Input (16 bits), multiplicand.

eY_B : Input (8 bits), multiplier.

load : Input - if high, loads the registers X and Y with eX_B and eY_B respectively in ph1.

4.1.2. Signals to/from Module on Chip

ph2I : Input, phase 2 inverted.

X_B2 : Output (16 bits) to mulmod_C (the MSB is sign bit).

Y_B2 : Output (8 bits) to mulmod_C (the MSB is sign bit).

ovrflw_1 : Input, 1 bit, overflow signal from the accmod_C (resets the inmod_C to zero).

4.2. Array Multiplier Module (mulmod_C)

This module calculates the final product (fp_B2) and sign (msign_2) of X_B2 and Y_B2. It uses the Wallace tree architecture. The two MSB from the inmod_C are fed into an XOR gate to give the sign bit, msign2, of the final product.

The multiplier module has four Wallace stages consisting of full adders (FA) and half adders (HA) and a Manchester Carry Lookahead Adder in the last stage. The table below shows the number of full and half adders used in the multiplier module.

Stage	Number of FA	Number of HA
0	27	4
1	14	1
2	13	0
3	13	2
Total	67	7

TABLE 1: FA and HA in the mulmod_C.

The last stage is a Manchester Carry Lookahead Adder and its operation is explained in section 3.4.3.

4.2.1. Signals to/from Module off Chip

ph1 : Input, phase 1 used to evaluate the Manchester adder.

4.2.2. Signals to/from Module on Chip

ph2I : Input, Inverted phase 2 used to "precharge" the Manchester adder circuit.

X_B2 : Input (16 bits), multiplicand, from inmod_C.

Y_B2 : Input (8 bits), multiplier, from inmod_C.

fp_B2 : Output (22 bits), final product of multiplier to tcamod_C.

msign_2 : Output (1 bit), sign bit, to tcamod_C.

4.2.3. Operation of the mulmod_C

The multiplier uses the Wallace tree architecture and a Manchester Carry Lookahead Adder in the last stage. A block diagram of the Multiplier and the Manchester Adder are shown in Fig 2.

4.3. Adder/Accumulator Module (accmod_C)

This module converts the final product output of the multiplier into the 2's complement format and adds it to the number stored in the accumulator (namely, register B).

4.3.1. Signals to/from Module off Chip

ph1 : Input, phase 1 clock.

ph1I : Input, inverted phase 1 clock.

ph2 : Input, phase 2 clock.

ph2I : Input, inverted phase 2 clock.

reset : Resets the accmod_C to zero.

4.3.2. Signals to/from Module on Chip

fp_B2 : Input (22 bits), final product from mulmod_C.

msign_2 : Input (1 bit), enables the one's complement circuitry.

ovrflw : Output (1 bit), high in case of an overflow.

sum : Output (23 bits), to outmod_C. The MSB is the sign bit.

4.3.3. Operation of accmod_C

The accmod_C module consists of various submodules, namely, the onecomp_C, the Areg23_C, Breg23_C, the Manadder_C and an sreg23_C.

The value of msign_2 serves as the control logic for the onecomp_C. If it is 1, it means that the number fp_B2 is negative and the 1's complement circuitry gets enabled. If it is zero it indicates that fp_B2 is positive and its 1's complement is the number itself. The output of the one's complement cell

(product) then gets added to the previous sum stored in the accumulator. We use a Carry Lookahead Manchester carry chain to implement the adder. A block diagram of the adder is shown in Fig 3.

The adder will add two numbers, the final product of the multiplier and the contents of the accumulator (i.e the previous sum). The numbers are 23 bit long, with the most significant bit being the sign bit. The two numbers are in the 2's complement format and the result of the adder will also be in the 2's complement format.

The carry C22 (generated out of the sum of the 22nd bits) and carry C23 (generated out of the sum of the 23rd bits) are of special interest. They specify if there is an overflow in the result (i.e result is incorrect).

The block diagram of the accmod_C is shown in Fig 4. with some examples of addition given in the appendix.

4.4. Output Module (outmod_C)

This module consists of the one's complement cell, the incr22_C and a dregB_C.

4.4.1. Signals to/from Module off Chip

ph1 : Input, phase 1 clock.

ph1I : Input, phase 1 inverted.

ph2 : Input, phase 2 clock.

ph2I : Input, phase 2 inverted.

rslt : Output (22 bits), the final product of the two input numbers.

signout: Output (1 bit), the sign of the final product.

ovrflwout : Output (1 bit), goes high in case of an overflow to indicate that the output result is incorrect.

reset : Input (1 bit), resets the dregB_C to zero.

4.4.2. Signals to/from Module on Chip

sum : Input (23 bits), from accmod_C (the MSB is the sign bit).

4.4.3. Operation of outmod_C

A block diagram of the outmod_C is shown in Fig 5. The onecomp_C takes the one's complement of the product from the accmod_C. The incr22_C increments the number by one, depending on whether the signin bit is zero or one. This final product in the sign-magnitude format is then stored inside the dregB_C which outputs the number in ph1.

5. Timing/Speed and Area

The target speed of the chip is 12 Mhz. The whole process of multilication right from the time the INMOD registers are loaded to the time the result is available at the output takes 3.5 cycles. The timing diagram of the operation of the MAC is shown in Fig 6.

The frame used for this chip was a 64 pin, 6900x6800 frame (64p69x68). The pin numbers with their corresponding names is given in Fig 7.

APPENDIX

SOME EXAMPLES :

Assume we are adding two numbers, each 5 bits long, with the MSB being the sign bit. All negative numbers are in 2's complement format.

Case 1:

```
  0 1 1 1 1        15
+ 0 1 1 0 1      + 13
0 1 1 1 0 0        28
```

C23= 0 C22=1 ==========> OVERFLOW (incorrect result)

Case 2:

```
  0 0 0 1 1        3
+ 0 1 0 0 1      + 9
0 0 1 1 0 0        12
```

C23=0 C22=0 ========> Result correct (01100= + 12)

Case 3:

```
  1 0 0 0 1       - 15
+ 0 0 1 1 0     +   6
0 1 0 1 1 1       - 9
```

C23=0 C22=0 ==========> Result correct (10111= -9)

Case 4:

```
  1 0 1 1 1       - 9
+ 1 1 1 0 0     + - 4
1 1 0 0 1 1       - 13
```

C23= 1 C22=1 ==========> Result correct (throw C23 and 10011= - 13)

FIG 1 BLOCK DIAGRAM OF MAC CHIP

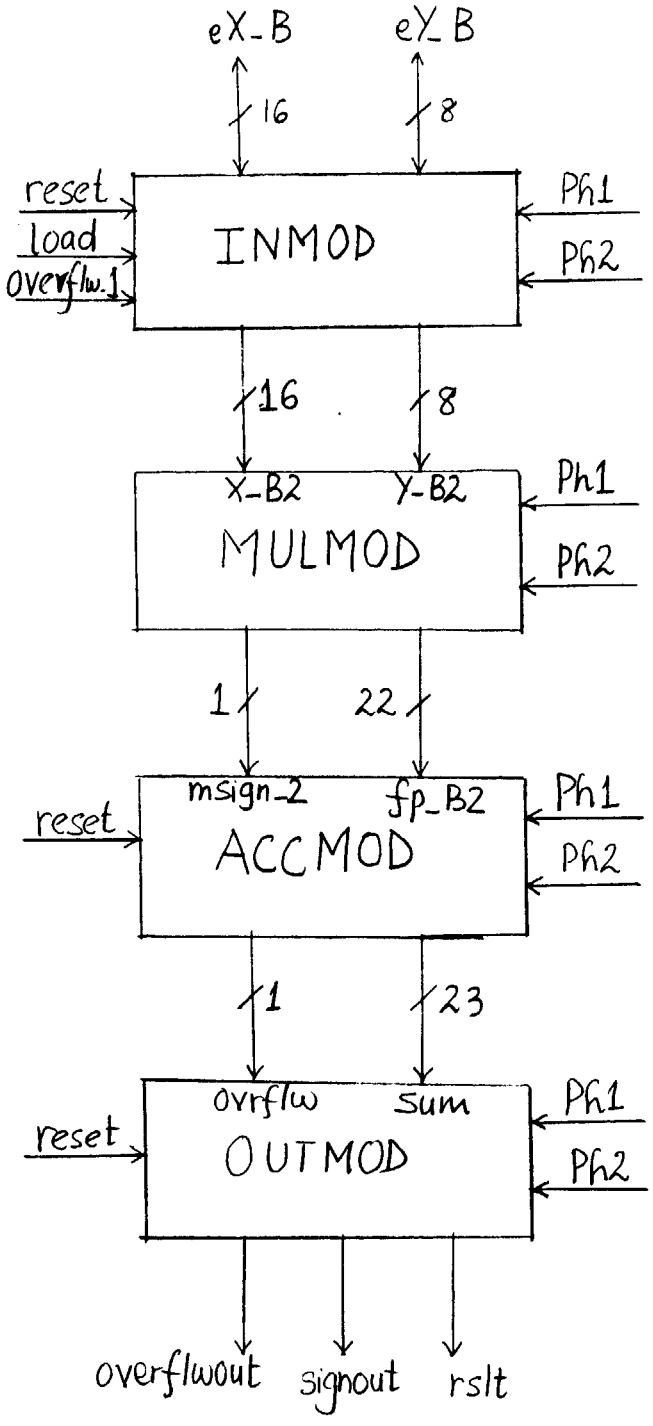
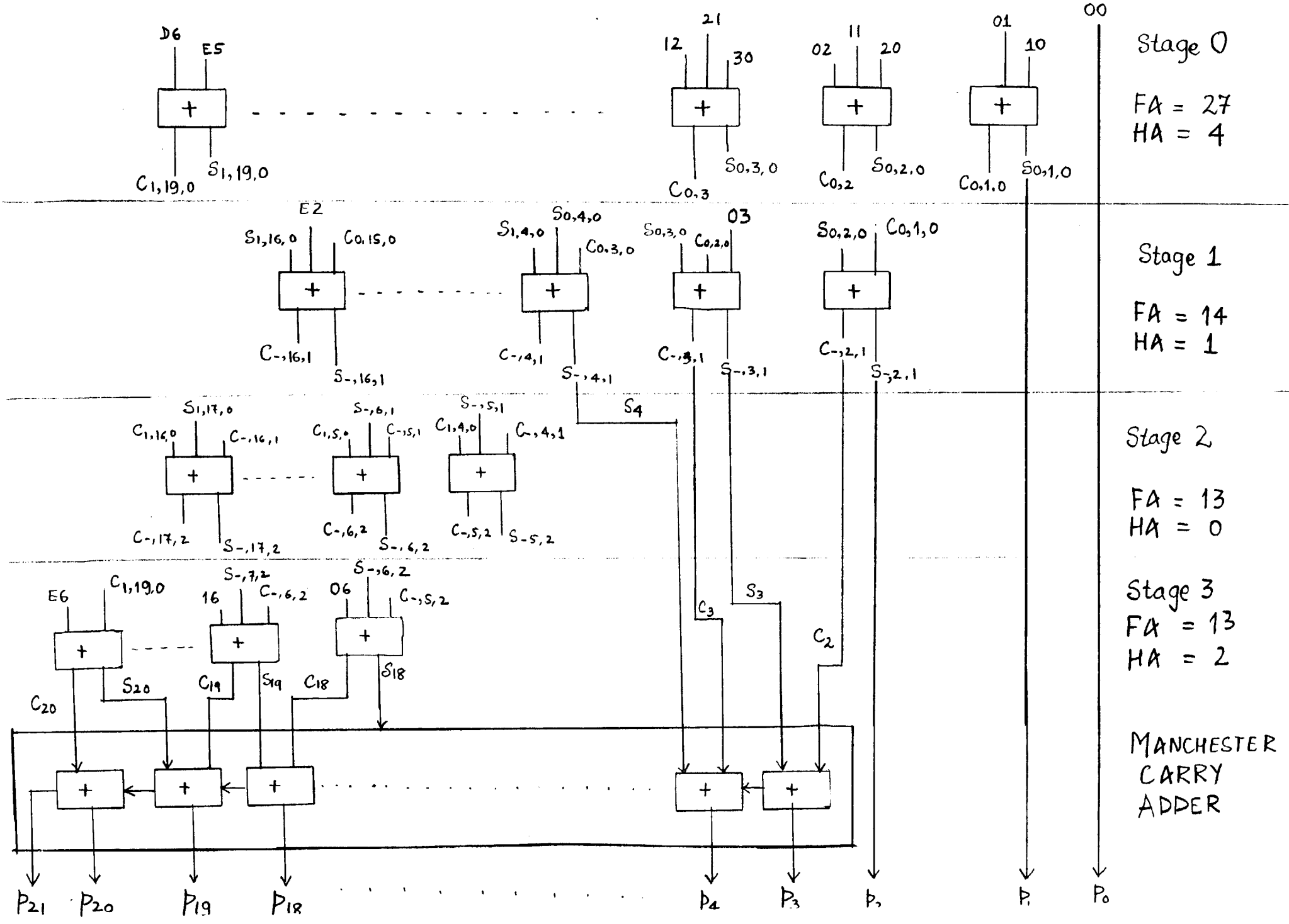


FIG 2(a) BLOCK DIAGRAM OF MULMOD



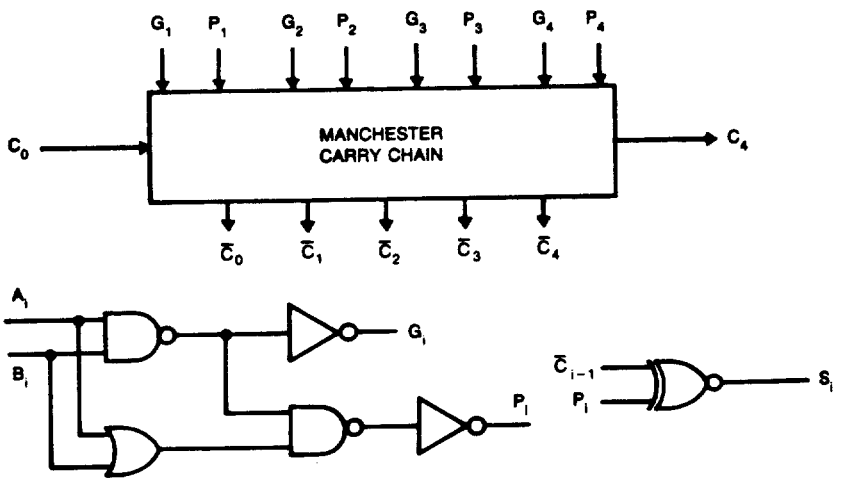


FIG 2(b) MANCHESTER CARRY ADDER

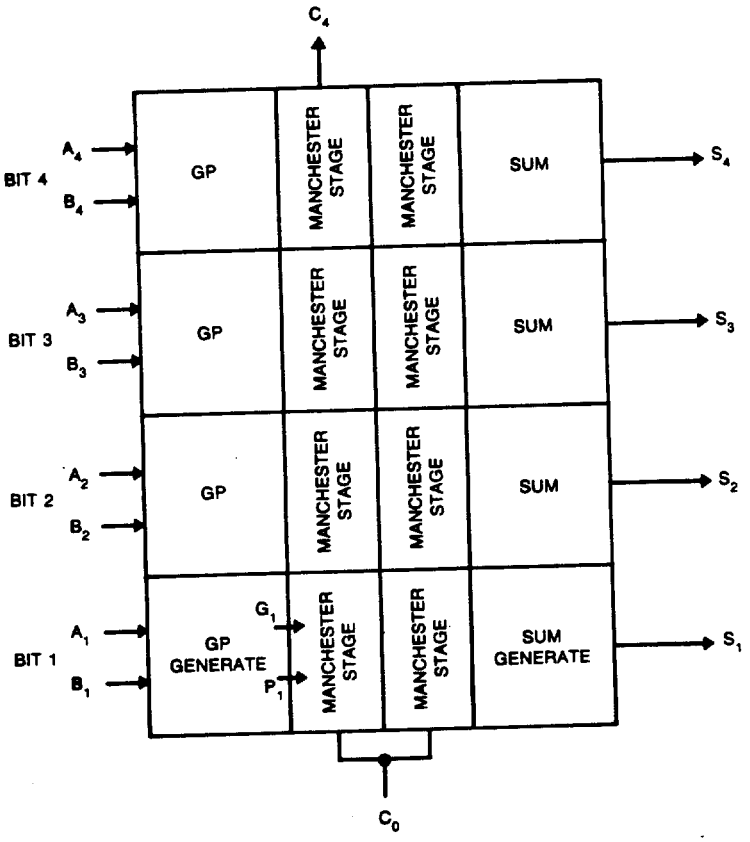


FIG 2(c): FLOOR PLAN FOR MANCHESTER CARRY ADDER

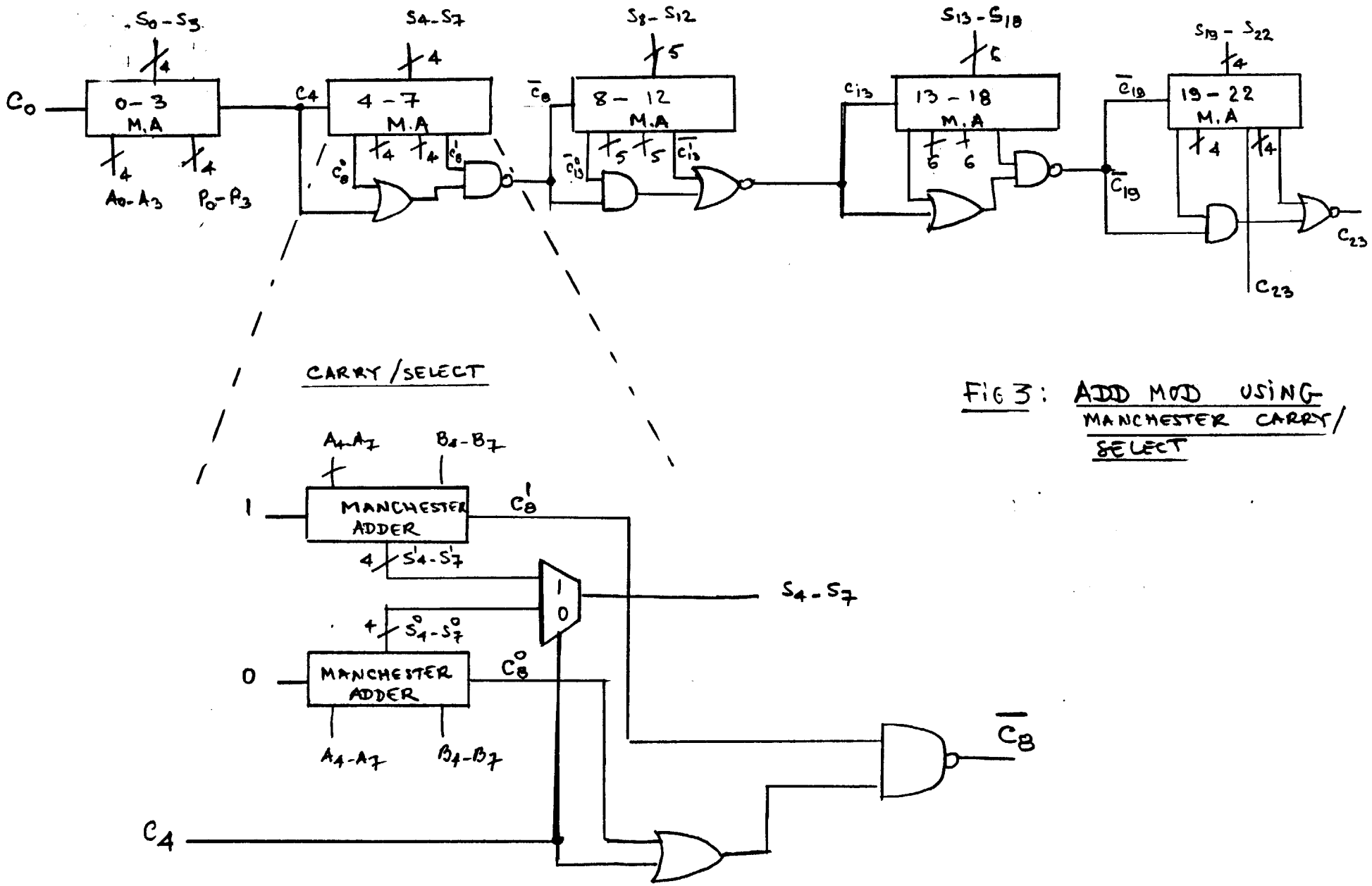


FIG 3: ADD MOD USING MANCHESTER CARRY/SELECT

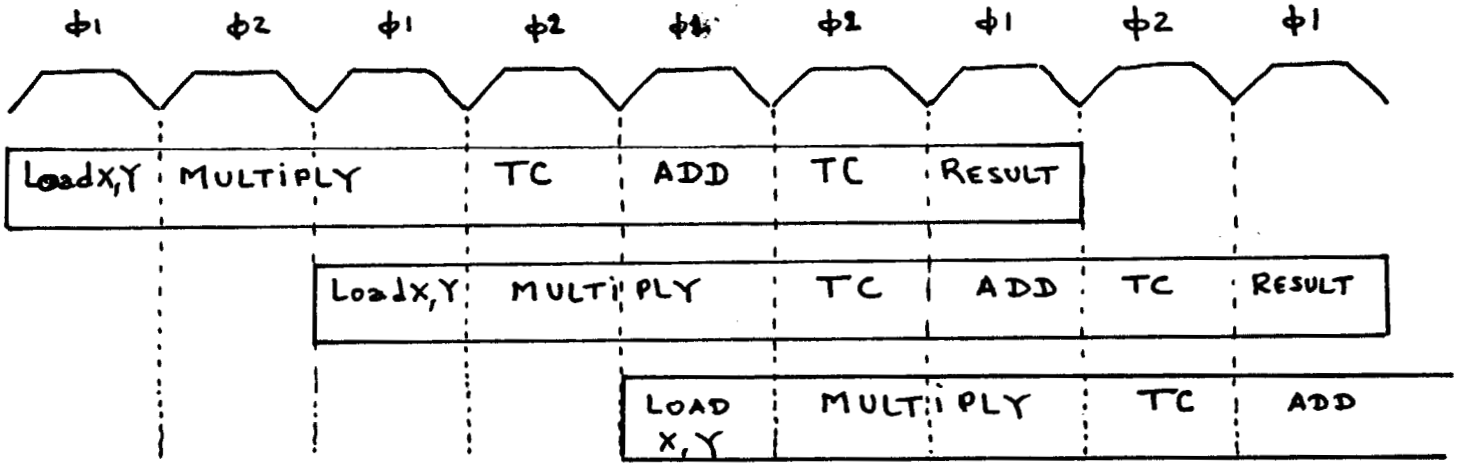
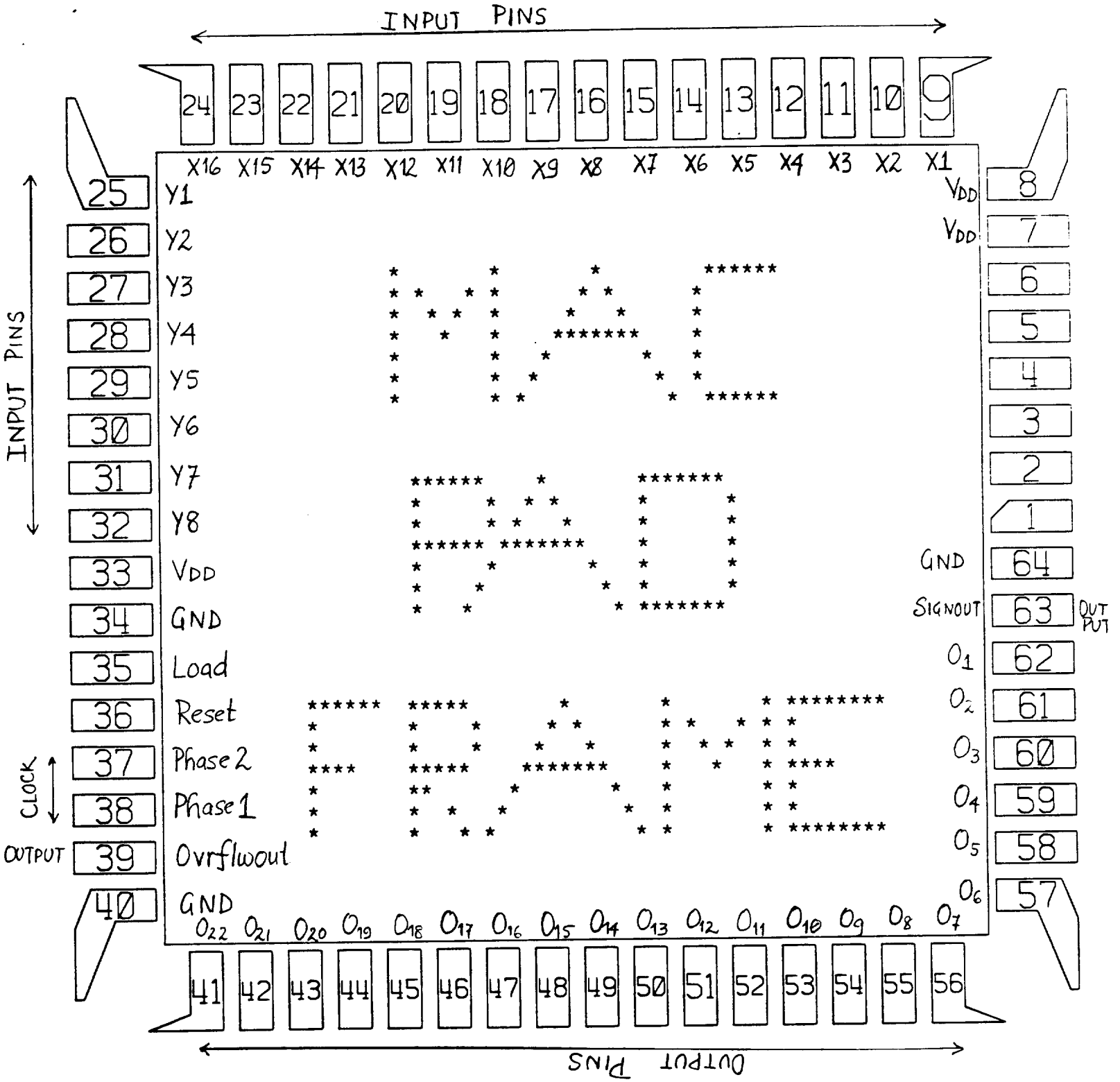


FIG 6: TIMING DIAGRAM

TC : 2's COMPLEMENT


INPUT PINS
24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9
X16 X15 X14 X13 X12 X11 X10 X9 X8 X7 X6 X5 X4 X3 X2 X1
25 Y1
26 Y2
27 Y3
28 Y4
29 Y5
30 Y6
31 Y7
32 Y8
33 VDD
34 GND
35 Load
36 Reset
37 Phase 2
38 Phase 1
39 Ovrflwout
40 GND
VDD 8
VDD 7
6
5
4
3
2
1
GND 64
SIGNOUT 63
O1 62
O2 61
O3 60
O4 59
O5 58
O6 57
INPUT PINS
CLOCK
OUTPUT
O22 O21 O20 O19 O18 O17 O16 O15 O14 O13 O12 O11 O10 O9 O8 O7
41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56
OUTPUT PINS

DIP64

FIG 7