AN ABSTRACT OF THE THESIS OF

Tsu-Ping Patrick Chuang for the Master of Science
(Name) (Degree)

Electric and
in Electronics Engineering presented on 3/15/72
(Major) (Date)

Title : SATELLITE MULTIPLICATION PACKAGE FOR A SMALL DIGITAL COMPUTER

Abstract approved : Redacted for privacy

James H. Herzog

This thesis is concerned with the design of an external multiplication package which can be utilized as an I/O device with a PDP-8/L computer. The multiplier and multiplicand are assumed to be 12 bit integers. The 24 bit product can be transferred back to the accumulator of computer 12 bits at a time. The control pulses for the operation are supplied by the computer through I/O transfer instructions.

The multiplication package was constructed on three printed-circuit cards, using only standard TTL IC chips. No other components were needed. It is simple, inexpensive and much faster than the method of repeated addition which must ordinarily be used in the PDP-8. This paper also shows a division algorithm using the multiplier and trial-and-error. This method of division is faster than repeated subtraction.

Satellite Multiplication Package
for A Small Digital Computer

by

Tsu-Ping Patrick Chuang

A THESIS

submitted to

Oregon State University

in partial fulfillment of

the requirements for the

degree of

Master of Science

June 1972

APPROVED:

Redacted for privacy

Associate Professor of Electrical and
Electronics Engineering

in charge of major

Redacted for privacy

Head of Department of Electrical and
Electronics Engineering

Redacted for privacy

Dean of Graduate School

Date thesis is presented 3/15/72

Typed by Won-Ho Chuang for Tsu-Ping Patrick Chuang

ACKNOWLEDGEMENT

The author wishes to express gratitude to Professor James H. Herzog for his initiation, advice and assistance in directing the design and writing of this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

SATELLITE MULTIPLICATION PACKAGE
FOR A SMALL DIGITAL COMPUTER

I. INTRODUCTION

The PDP-8/L small digital computer uses 12 bit words in its accumulator and memory. Its memory cycle time is 1.6 us. There are six memory reference instruction, one of them is an add instruction called " Two's Complement Add " (TAD). The multiplication ordinarily has to be done by repeated addition of the multiplicand. There are also problems with overflow. A 12-bit by 12-bit multiplication using this method has to use a double precision procedure in order to get the 24-bit product. It is obviously time consuming. A hardware 12-bit by 12-bit multiplication using the existing interface utility of PDP-8/L is very helpful to save programming time.

This hardware multiplication package is called " satellite ", because it is an external package which can be plugged into I/O rack of the PDP-8/L when needed.

A reasonable algorithm to consider for an external package is a shift and add algorithm. For a 12 bit multiplier, only 12 additions are required.

Here is an example utilizing a 4 bit multiplicand and 4 bit multiplier. The multiplicand is 13 (=1101),the multiplier is 11 (=1011). The result should be 143 (=10001111).

Comment	Partial sum	Multiplier
	Overflow	Test
Start	0 0 0 0 0	(1 0 1 1
Test/Add	0 1 1 0 1	(1 0 1 1
Shift Right	0 0 1 1 0	1(1 0 1
Test/Add	1 0 0 1 1	1(1 0 1
Shift Right	0 1 0 0 1	1 1(1 0
Test/Add	0 1 0 0 1	1 1(1 0
Shift Right	0 0 1 0 0	1 1 1(1
Test/Add	1 0 0 0 1	1 1 1(1
Shift Right	0 1 0 0 0	1 1 1 1

We start from zero partial sum. The " Test/Add " means that the last bit of multiplier is tested. If it is "1", then add the multiplicand to partial sum. If it is "0", then add nothing to partial sum. The " Shift Right " operation should shift both the partial sum and the multiplier, and drop the last bit (which was tested in the previous step) of multiplier. Repeat the " Test/Add " and " Shift Right " operations until every bit of the multiplier has been used. The most significant bits of product will be in the partial sum. The least significant bits of product will be in the original position of the multiplier.

Following the above algorithm, the logic circuit design will be in Chapter II. Chapter III talks about the interface and generation of enable pulses. Chapter IV will supply a multiplication sample program and a division sample program using this hardware multiplication. Detailed circuit design will be supplied in the appendix.

II. LOGIC DESIGN OF MULTIPLICATION PACKAGE

According to the multiplication algorithm in the previous chapter, there must be three registers (each 12 bits in length) to hold the multiplicand, multiplier and partial sum. These will be called R, MQ and A register respectively. A hardware adder is also needed. The simplified register layout is shown on Fig.1.

Register Layout and Information Flow

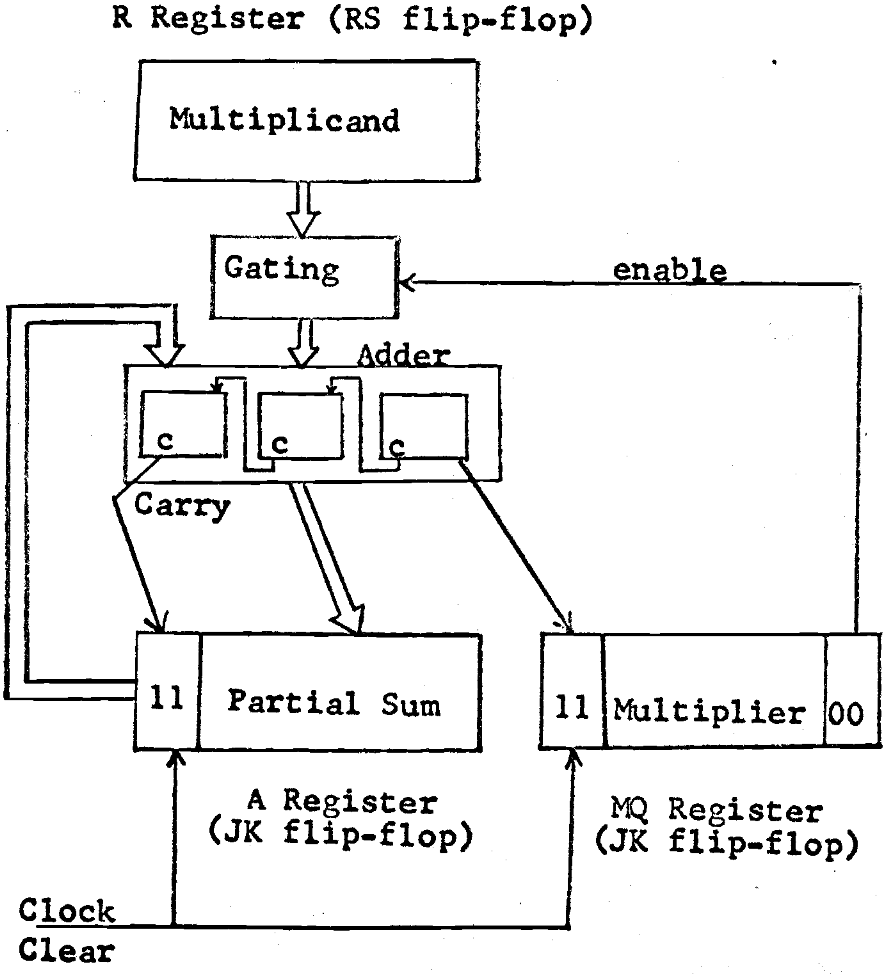


Fig.1. Logic block diagram.

Procedure

(1) Clear all registers. Load multiplier into accumulator of PDP-8/L.

(2) From accumulator, transfer multiplier to R register first, then to MQ register. (R register is the only channel to accept the information coming from accumulator of PDP-8/L.)

(3) Load multiplicand into accumulator, and from there transfer to R register.

(4) Perform the Test-Add-Shift Right operation. This is done by hardware. The last bit of multiplier enables the addition of the R and A register. The 13-bit sum is sent to A register and MQ11 as shown. The "Shift Right" is done by actual circuit connections.
A clock pulse is needed to transfer the information in A and MQ register from the JK master side to Q$\overline{Q}$ slave side.

(5) Repeat step 4, 12 times (or simply say, 12 clock pulses).

(6) Return the twelve most significant bits of product in A register to accumulator.

(7) Return the twelve least significant bits of product in MQ register to accumulator.

Implementation

(1) R Register

R register uses RS flip-flops which are made from two 2-input NAND gates (type SN7400N) as shown in Fig. 2. It is loaded from the corresponding bufferred accumulator bit of PDP-8/L. Its output is sent to the MQ register or to the adder.

RS flip-flop truth table

S	R	Q(t+1)
0	0	?
0	1	1
1	0	0
1	1	Q(t)

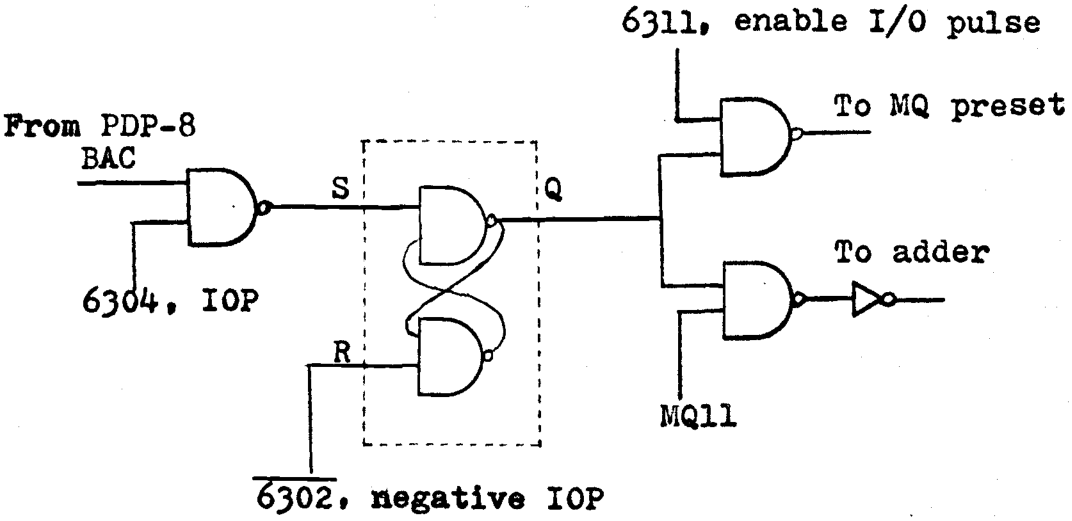


Fig. 2. R register, multiplicand bit.

(2) A Register

Register A uses JK master-slave flip-flops (SN7476N). The typical stage is shown on Fig. 3. It is loaded from the sum bits of the adder. There are thirteen sum bits in the output of adder. Only the twelve most significant bits are loaded into A register. The " Shift Right " operation is done by this kind of actual connections.

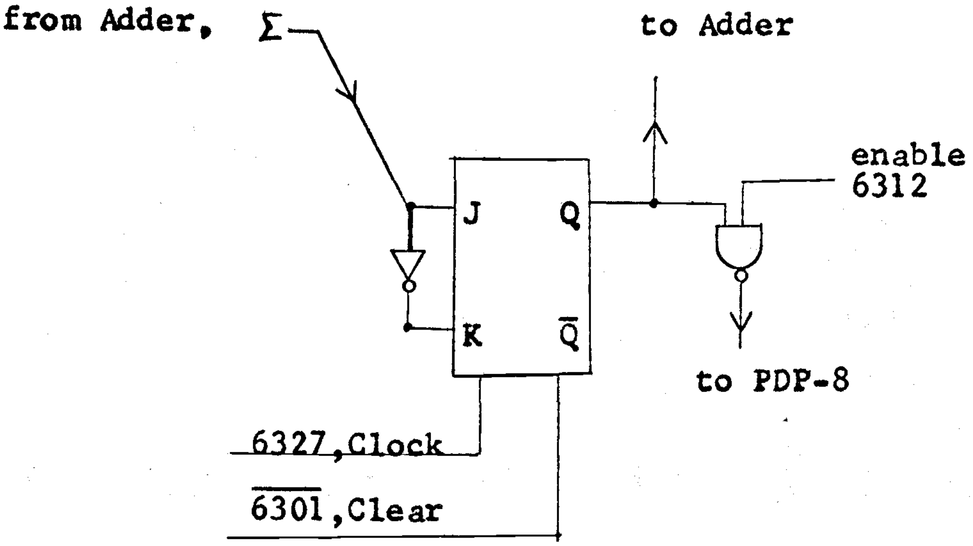


Fig. 3. Register A, partial sum bit.

(3) MQ Register

Register MQ also uses JK flip-flops (SN7476N). The typical stage is shown on Fig. 4. It is loaded from the R register. Its output is sent to the master side of next stage or to the corresponding bit of accumulator.

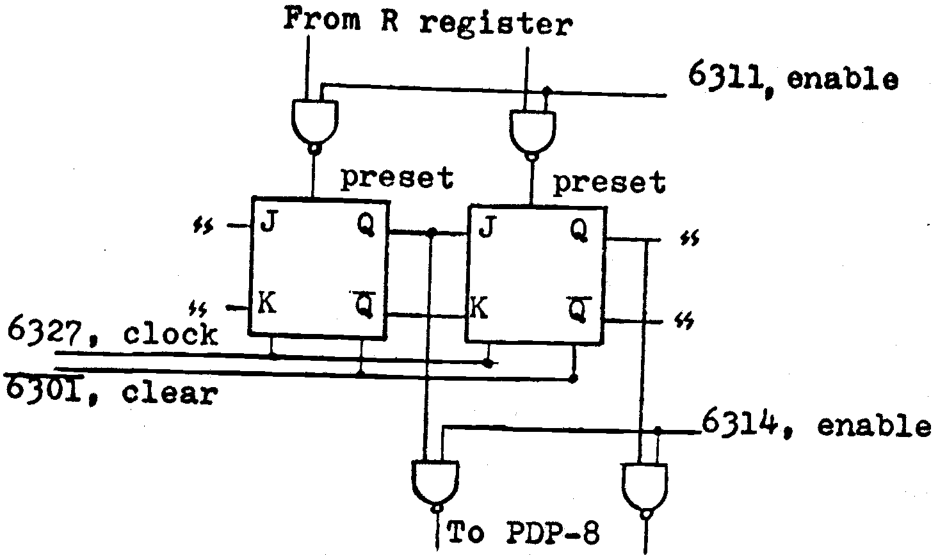


Fig. 4. MQ register, multiplier bit.

(4) Adder

The adder uses three 4-bit full adders (SN7483N). It adds the R and A register. The 13-bit sum is sent to the A register and MQ11, as mentioned before.

III. INTERFACE WITH PDP-8/L AND GENERATION OF ENABLE PULSES

Programmed Transfer

There are three basic methods for the transfer of information between I/O devices and the PDP-8. The method used here is called "Programmed Transfer", in which instructions are included at some point in the program to accept or transmit information. Thus, programmed transfers are program initiated and are under program control. This method uses the accumulator as the buffer, or storage area in the computer, for all data transfers.

The octal operation code "6" is used to specify an input/output transfer (IOT) instruction. The typical octal IOT instruction is like the one in Fig. 5.

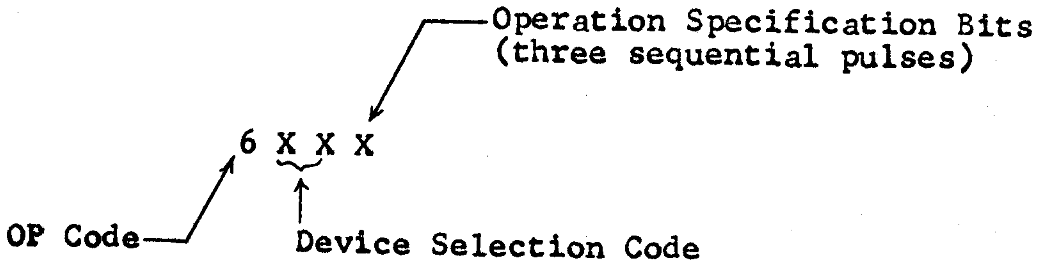


Fig. 5. The octal IOT instruction

Device Selection and I/O Pulses

The device selection code is transmitted to all peripheral equipment (like the Teletype and this hardware multiplier) whenever the IOT instruction is executed. Each IOT instruction may generate as many as three sequential pulses (each 0.8 us in length, seperated by 0.1 us. Total I/O instruction time is 4.25 us).

When executing the I/O instruction, one or two or all of the three IOP terminals (see appendix A, I/O rack) will generate positive pulse. For example, when executing "6XX1", only the "IOP 1" terminal will generate a positive pulse (logic 1). There is no pulse generated from "IOP 2" or "IOP 4" terminal. Same for "6XX2" (only "IOP 2" terminal will be logic 1) and "6XX4" (only "IOP 4" terminal will be logic 1). But when executing "6XX7", all of the three IOP terminals will generate positive pulse. When executing "6XX6", both "IOP 2" and "IOP 4" terminal will generate positive pulse.

These three I/O pulses are further gated by a device selector (a 6-input NAND gate). Figure 6 shows the details for generating nine enable pulses associatdd with device selectors 30,31 and 32 (octal value). Some of these enable pulses are negative, as required to clear and preset the JK flip-flops.

Required IOT instruction

The following IOT instructions are used to perform multiplication. Three octal device codes (30, 31 and 32) are used. Where 6307 and 6306 are combination instructions.

Proposed New Mnemonic Code	IOT Inst.	Comments (positive or negative pulse needed)
CAM	6301	Clear A and MQ register (-)
CRR	6302	Clear R register (-)
TFR	6304	Transfer contents of accumulator of PDP-8/L (AC) into R (+)
CRI	6307	Clear MQ and R, transfer AC into R (a combination of CAM, CRR and TFR)
RIN	6306	Clear R and transfer AC into R (a combination of CRR and TFR)
MQI	6311	Transfer contents of R into MQ (+)
AOT	6312	Transfer contents of A register to PDP-8/L accumulator (+)
MQO	6314	Transfer contents of MQ register to PDP-8/L accumulator (+)
CLK	6327	Supply three clock pulses to A and MQ register (+)

Interface Logic Design Note

The output circuitry from the PDP-8 contains TTL IC chips, 7400 series. Their standard fan-out is ten loads. In order to save some fan-out capability to other external package, it is better take only one load from each output terminal (see appendix A) to this package. The circuit shown on Fig. 6 has taken care of this.

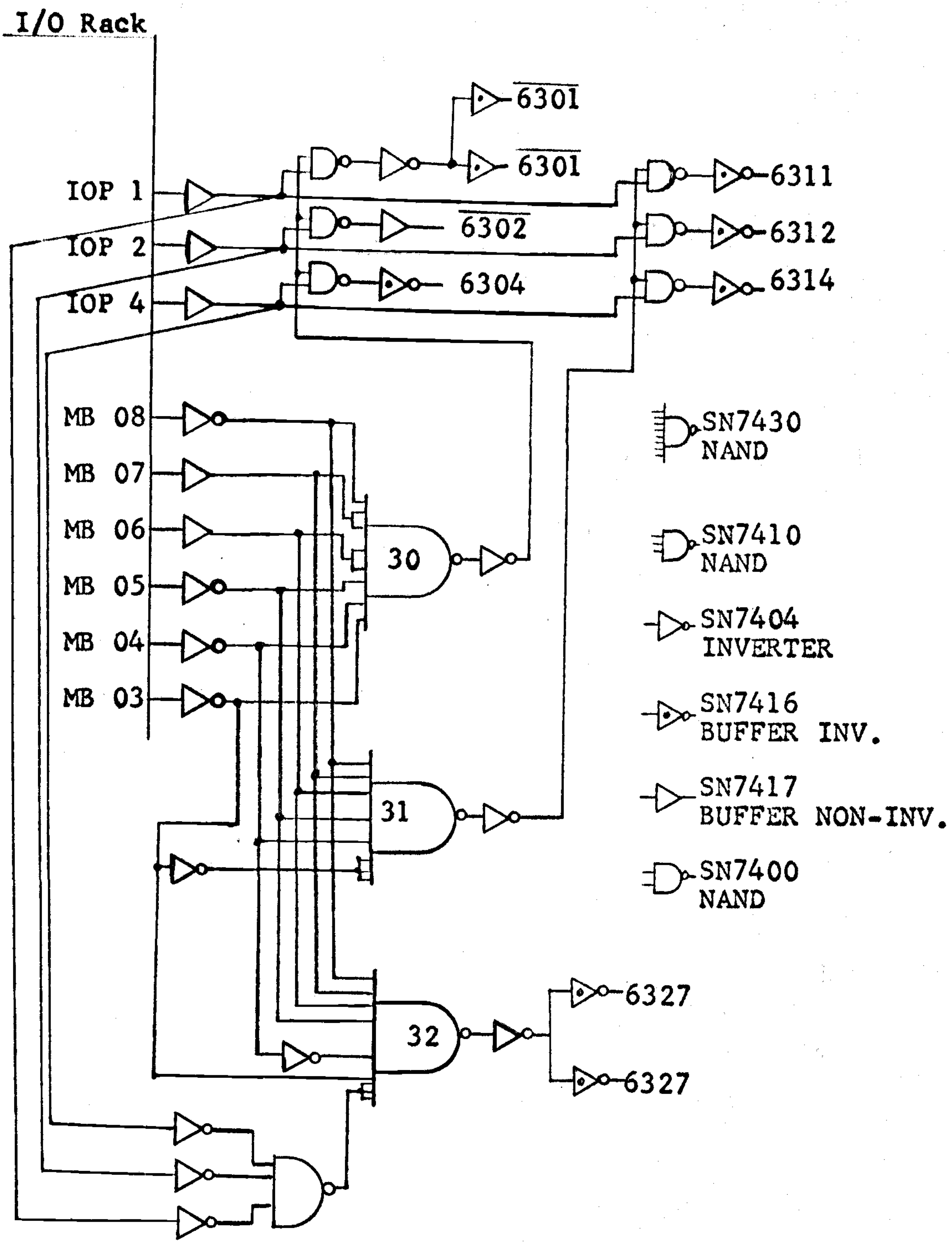


Fig. 6. Generation of enable pulses
through device selectors

IV. RESULTS

A 12-bit by 12-bit hardware multiplier has been implemented using fifty-five IC chips on three printed-circuit cards. Each card is 5.5 inches by 4.5 inches. The total cost of IC chips is around $30.00.

This package plugs directly into the I/O rack of the PDP-8/L and operates under program transfer mode. A program or subroutine is required.

The associated program for multiplication is shown on the next page where a sample octal machine language program was actually run and the whole program was cut from the paper of Teletype printer. The programmer using this package has to follow the basic program in order to get the product. But he can arrange the program so that his own program job can be done more efficiently.

The list of IC chips and a sample division program using the multiplication package are shown on following pages.

The Sample Program for Multiplication

```
Symbolic Inst,          Comments

*200
START, CLA CLL          Clear AC
       TAD MQ           Add multiplier to AC
       CRI              Clear A, MQ, R and
                        transfer AC to R
       MQI              Transfer R to MQ
       CLA CLL          Clear AC
       TAD RA           Add multiplicand
                        to AC
       RIN              Clear R and transfer
                        AC to R
       CLK  )
       CLK  |           3 clock pulses each
       CLK  |           instruction, total
       CLK  )           12 clock pulses
       CLA CLL          Clear AC
       AOT              A register transfers
                        to AC
       DCA RA           Store AC in RA
                        location
       MQO              MQ register transfers
                        to AC
       DCA MQ           Store AC in MQ
                        location
       HLD              Hold
RA,    xxxx             Data of multiplicand,
                        then contents of A
MQ,    xxxx             Data of multiplier,
                        then contents of MQ
$
```

ODT octal Program

```
200/7300
0201 /1222
0202 /6307
0203 /6311
0204 /7300
0205 /1221
0206 /6306
0207 /6327
0210 /6327
0211 /6327
0212 /6327
0213 /7300
0214 /6312
0215 /3221
0216 /6314
0217 /3222
0220 /7402
0221 /7777
0222 /7777

200G

0221/7776
0222 /0001

200G

0221/0000
0222 /7776

200G

0221/0000
0222 /0000
```

(Remark: Use the proposed new mnemonic codes.)

The IC chips used in this package are shown as below:

T. I. IC series	Number of chips	Approximate price
SN7400N	25	$ 0.20 ea.
SN7404N	6	$ 0.20 ea.
SN7405N	2	$ 0.30 ea.
SN7410N	1	$ 0.20 ea.
SN7416N	2	$ 0.20 ea.
SN7417N	1	$ 0.20 ea.
SN7430N	3	$.0.20 ea.
SN7476N	12	$ 1.00 ea.
SN7483N	3	$ 3.00 ea.
Total :	55 chips	$29.20

A Sample Program for Division by Using Multiplication Package

The integer division C/D=Q+R/D, where C≥D, can be done by the following trial-and-error method. The maximum octal value for quotient Q is 7777. So let's try Q=100,000,000,000 (i.e., assign the possible most significant bit to 1). If the product of QxD is smaller than or equal to the dividend C, then the "1" stays in that bit position. Otherwise change the "1" to a "0". Then try the next most significant bit to see if it is a binary "1" (i.e., try Q=X10,000,000,000 where X has been found by previous trial), and so on, up to the last bit. After twelve trials, the quotient Q is found and the remainder R=C-QxD.

The flow chart and symbolic language program are shown on following pages. A glossary is supplied here for better understanding.

Glossary

C:	dividend
D:	divisor
Q:	quotient
R:	remainder
CTC:	Two's Complement of C
TRY:	first value to try(here is 4000_8)
K:	variable trial value
$\overline{K}$:	one's complement of K variable
A:	most significant part of product QxD, a 12-bit number
MQ:	least significant part of product QxD, a 12-bit number

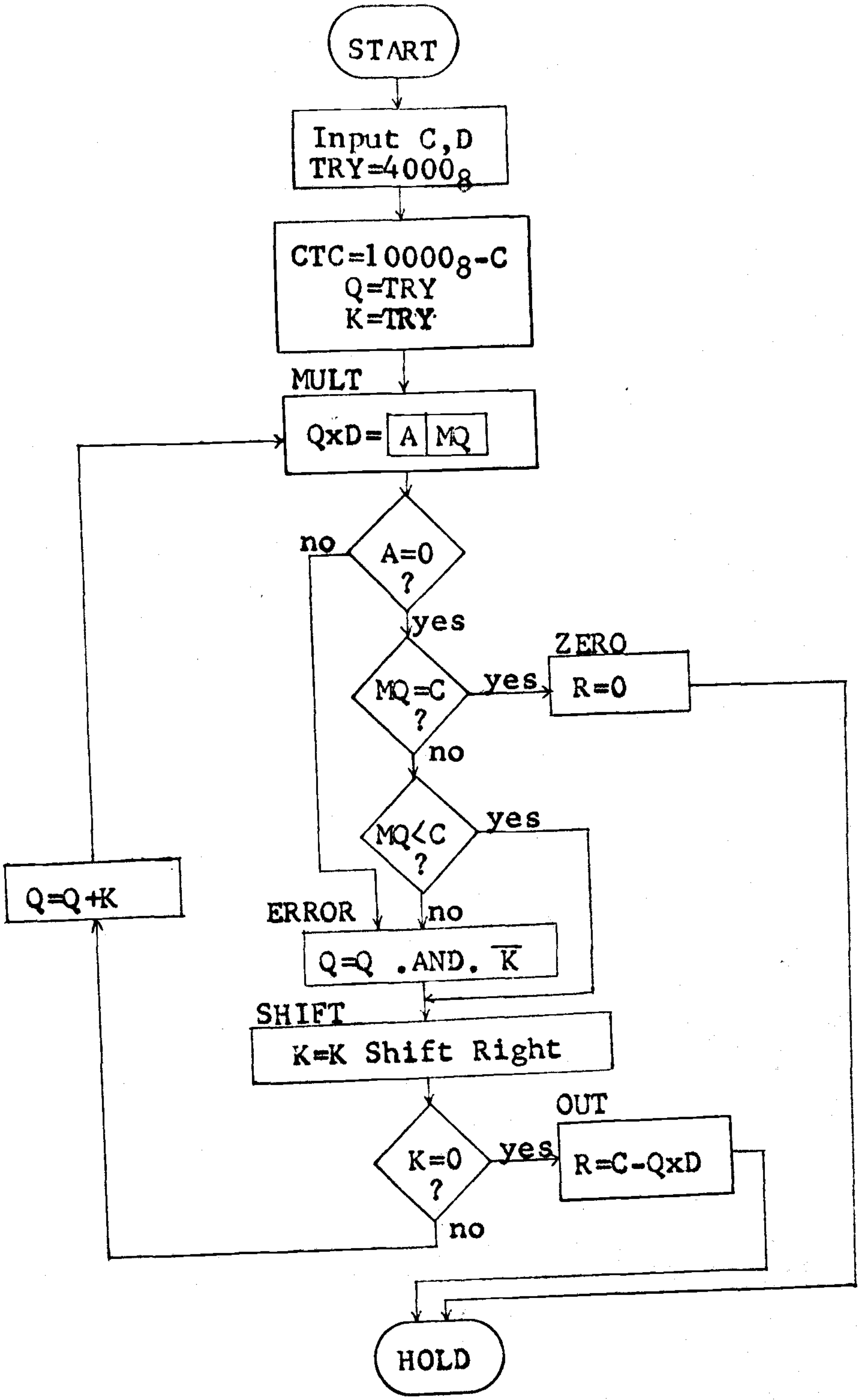


Fig. 7. Flow chart for division

Symbolic Program for Division

```
*200
START,    CLA CLL
          TAD C
          CIA
          DCA CTC
          TAD TRY
          DCA K
          TAD TRY
          DCA Q
MULT,     JMS MULTY
          TAD A
          SZA
          JMP ERROR
          TAD MQ
          SPA
          JMP ERROR
          TAD CTC
          SNA
          JMP ZERO
          SPA
          JMP SHIFT
ERROR,    CLA CLL
          TAD K
          CMA
          AND Q
          DCA Q
SHIFT,    CLA CLL
          TAD K
          RAR
          SNA
          JMP OUT
          DCA K
          TAD K
          TAD Q
          DCA Q
          JMP MULT
ZERO,     DCA R
          JMP END
OUT,      JMS MULTY
          TAD MQ
          CIA
          TAD C
          DCA R
END,      HLD
```

```
C,        XXXX
D,        XXXX
Q,        0000
R,        0000
CTC,      0000
TRY,      4000
K,        0000
A,        0000
MQ,       0000



MULTY,    0000
          TAD Q
          IOT+307
          IOT+311
          CLA CLL
          TAD D
          IOT+306
          IOT+327
          IOT+327
          IOT+327
          IOT+327
          CLA CLL
          IOT+312
          DCA A
          IOT+314
          DCA MQ
          JMP I MULTY
$
```

(Remark: $0 \leq C \leq 3777_8$.)

V. CONCLUSION

The 12-bit by 12-bit multiplication package is used under program control of PDP-8. When we look at the sample program for multiplication previously supplied, we can see that the total operation time is 115.35 us (or about 32 memory cycle times.).

For the division program which uses the hardware multiplier, the maximum program time required is about 800 memory cycle times. The average program time will be about 700 memory cycle times. It is faster than the usual repeated subtraction routine method which requires 700 memory cycle times when the quotient is around 53 (decimal). When the quotient is greater than 53, more than 700 memory cycle times will be needed.

It is possible to modify this package in hardware, in order to do the hardware division. Mainly, a one bit operation code register, hardware compare gating circuit and shift left operation gating circuit will be added.

Supplied in appendix B, the detailed circuit design and the card/chips layout are very helpful if something is wrong in this package.

BIBLIOGRAPHY

1. Digital Equipment Corporation. Small computer handbook. Maynard, Massachusetts, 1970.

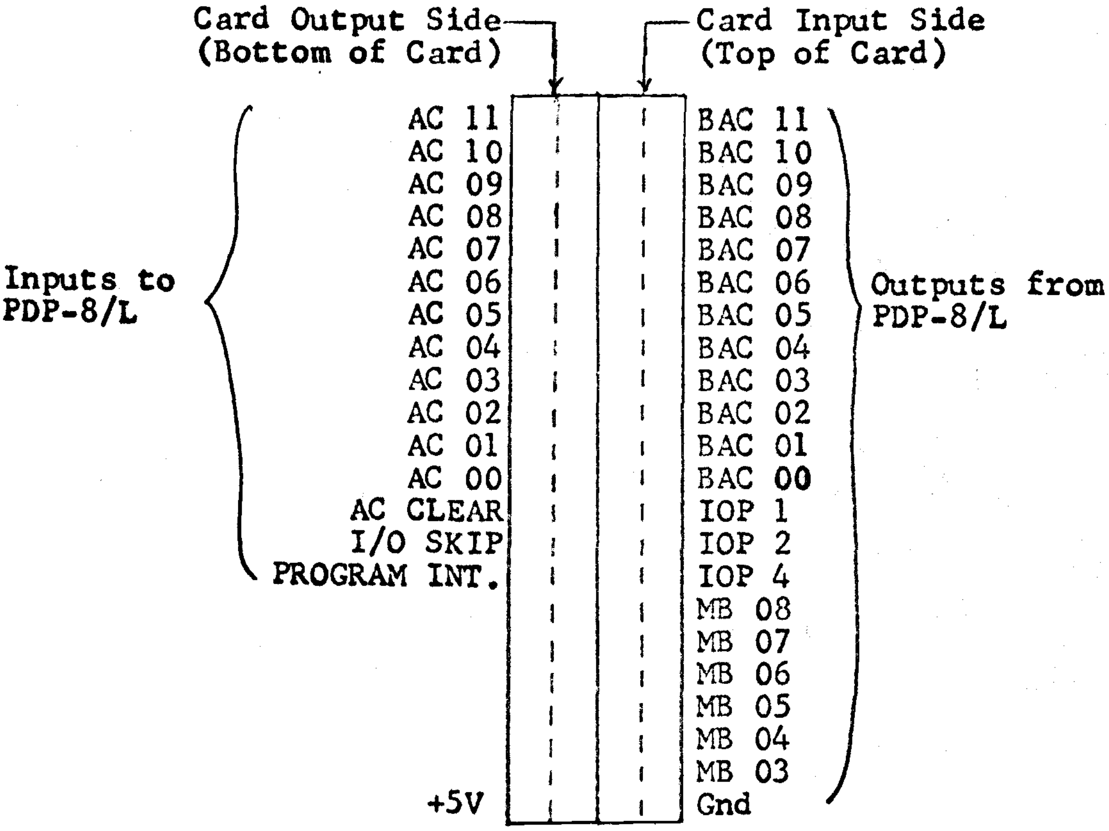
2. Digital Equipment Corporation. Introduction to programming. PDP-8 Family. Maynard, Massachusetts, 1970.

3. Texas Instruments Incorporated. The Integrated Circuits Catalog for Design Engineers. 1971.

APPENDICES

APPENDIX A: I/O RACK TERMINALS

FRONT VIEW (Card plug-in side)



Card Output Side (Bottom of Card)	Card Input Side (Top of Card)
AC 11	BAC 11
AC 10	BAC 10
AC 09	BAC 09
AC 08	BAC 08
AC 07	BAC 07
AC 06	BAC 06
AC 05	BAC 05
AC 04	BAC 04
AC 03	BAC 03
AC 02	BAC 02
AC 01	BAC 01
AC 00	BAC 00
AC CLEAR	IOP 1
I/O SKIP	IOP 2
PROGRAM INT.	IOP 4
	MB 08
	MB 07
	MB 06
	MB 05
	MB 04
	MB 03
+5V	Gnd

NOTE: 11=most significant bit (equal to bit position assignment 00)

00= least significant bit (equal to bit position assignment 11)

Outputs from PDP-8/L are positive logic.
Inputs to PDP-8/L are negative logic.

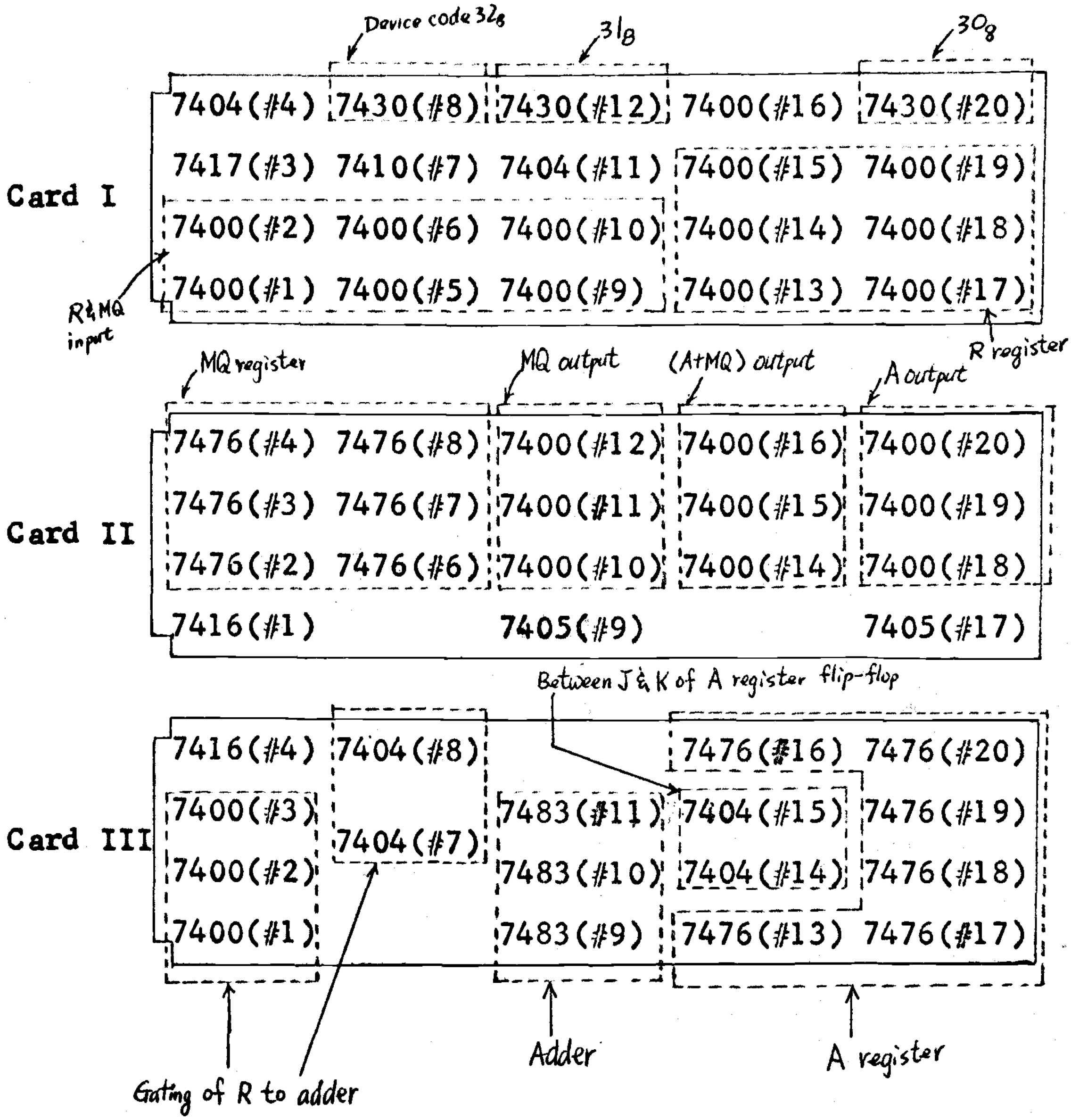
All gates are Texas Instruments Series 7400 TTL

Logic "1" ≥ 2.4 volts
Logic "0" ≤ 0.4 volts

APPENDIX B

THE CARD/CHIP LAYOUT AND DETAILED CIRCUIT DESIGN

There are three cards, namely Card I, Card II and Card III. Each card has twenty chip positions. In the diagram blow, the type of chip follows by a position number.



Card I

7404(#4)	7430(#8)	7430(#12)	7400(#16)	7430(#20)
7417(#3)	7410(#7)	7404(#11)	7400(#15)	7400(#19)
7400(#2)	7400(#6)	7400(#10)	7400(#14)	7400(#18)
7400(#1)	7400(#5)	7400(#9)	7400(#13)	7400(#17)

Card II

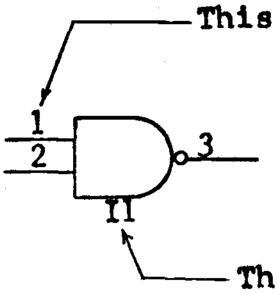
7476(#4)	7476(#8)	7400(#12)	7400(#16)	7400(#20)
7476(#3)	7476(#7)	7400(#11)	7400(#15)	7400(#19)
7476(#2)	7476(#6)	7400(#10)	7400(#14)	7400(#18)
7416(#1)		7405(#9)		7405(#17)

Card III

7416(#4)	7404(#8)		7476(#16)	7476(#20)
7400(#3)	7404(#7)	7483(#11)	7404(#15)	7476(#19)
7400(#2)		7483(#10)	7404(#14)	7476(#18)
7400(#1)		7483(#9)	7476(#13)	7476(#17)

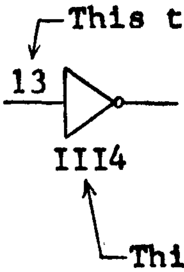
In order to pin-point the terminals of the chips from the detailed circuit diagram, here are some examples.

This terminal is at 1st pin of chip I1.



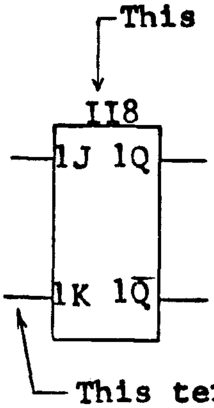
This NAND gate is on Card I, #1 chip.

This terminal is at 13th pin of chip III4.



This inverter is on Card III, #4 chip.

This JK flip-flop is on Card II, #8 chip.



This terminal is at 1K pin.

Fig. 8. Detailed circuit diagram (I)

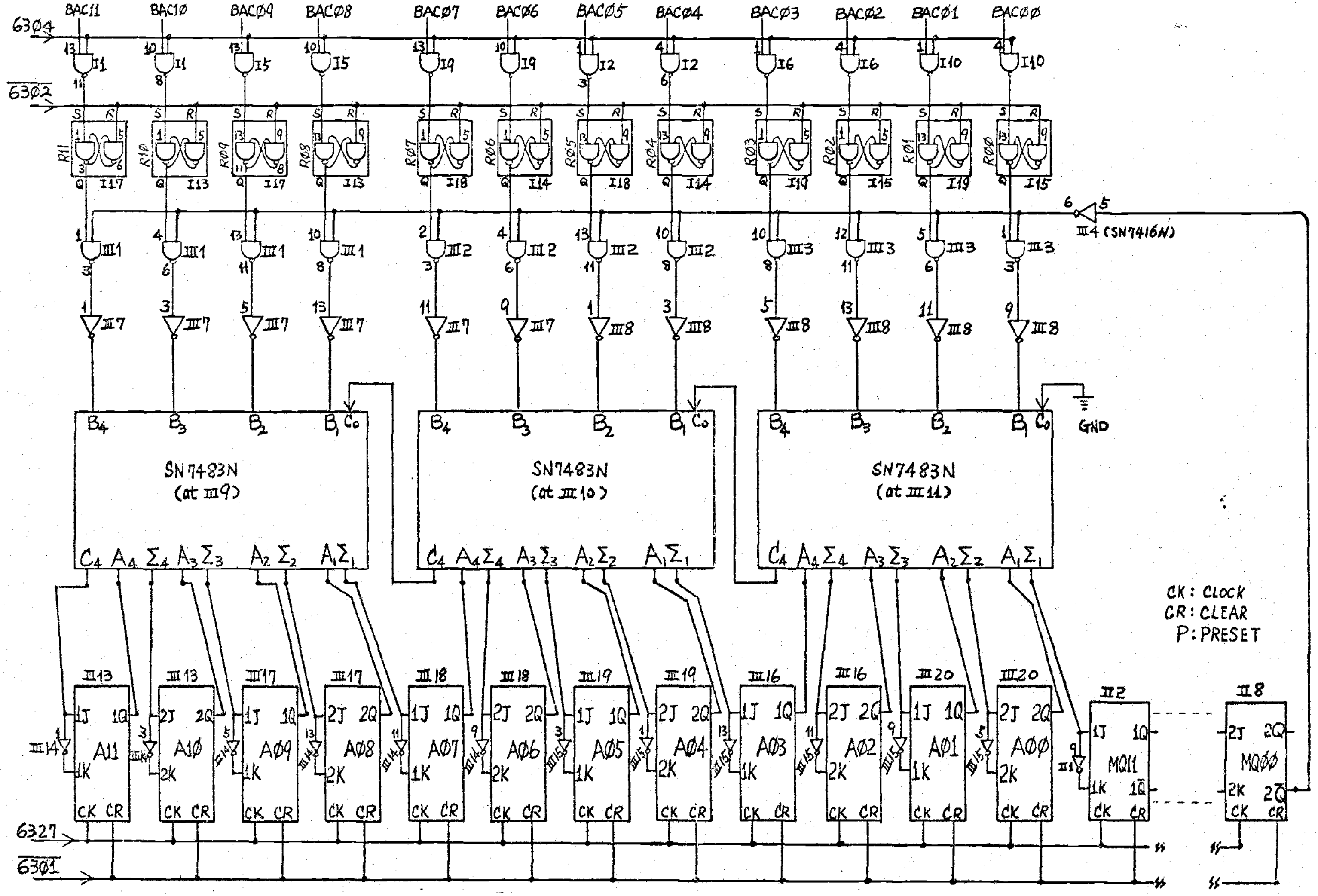
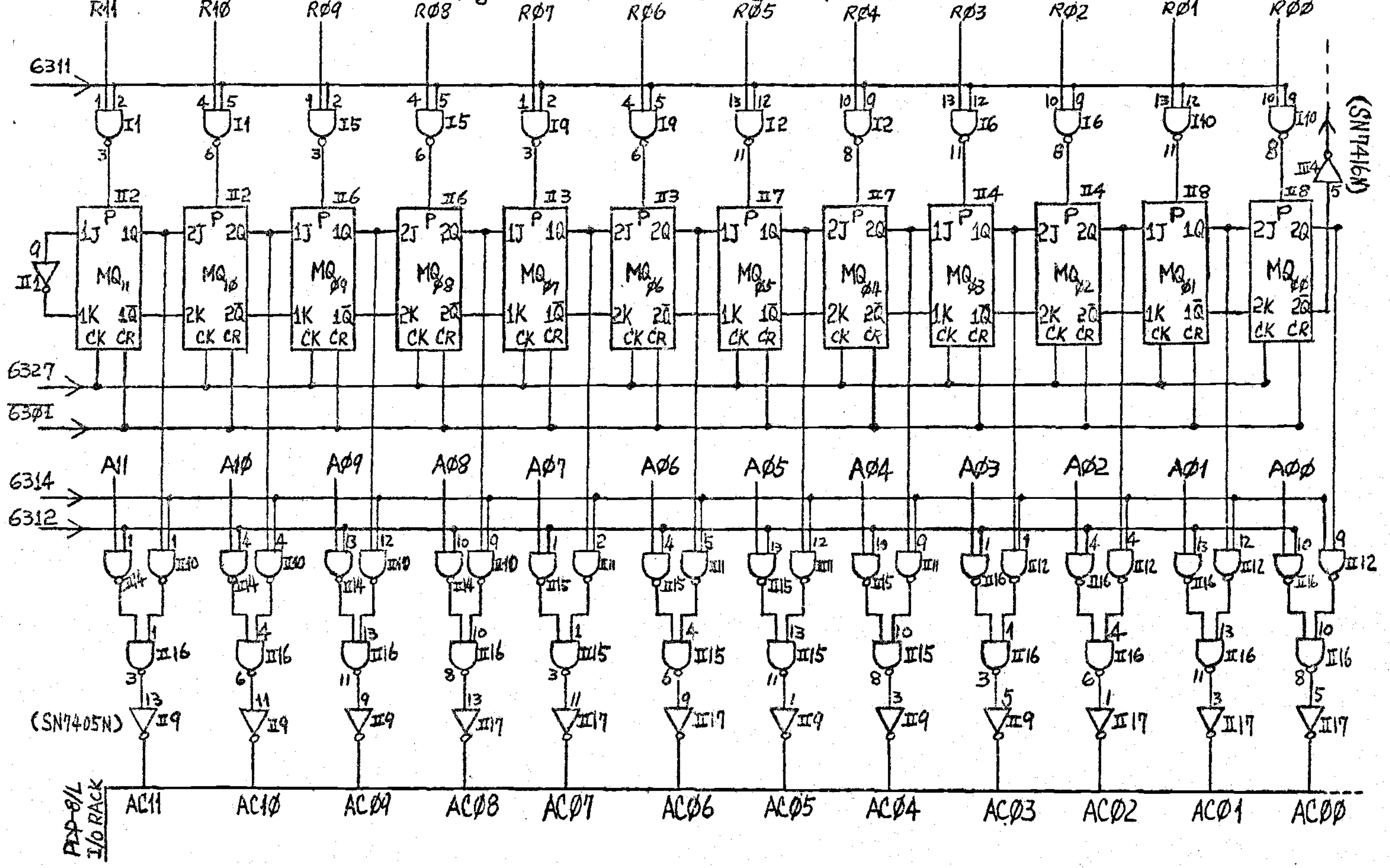


Fig. 9. Detailed circuit diagram (II)



R11
R10
RØ9
RØ8
RØ7
RØ6
RØ5
RØ4
RØ3
RØ2
RØ1
RØØ
6311
I1
I5
I9
I2
I6
I10
(SN7416N)
II4
II2
II6
II3
II7
II4
II8
II1
P
1J
1Q
1K
1Q̄
2J
2Q
2K
2Q̄
CK
CR
MQ11
MQ10
MQØ9
MQØ8
MQØ7
MQØ6
MQØ5
MQØ4
MQØ3
MQØ2
MQØ1
MQØØ
6327
6301
6314
6312
A11
A10
AØ9
AØ8
AØ7
AØ6
AØ5
AØ4
AØ3
AØ2
AØ1
AØØ
II14
II10
II15
II11
II16
II12
(SN7405N)
II9
II17
PDP-8/L
I/O RACK
AC11
AC10
ACØ9
ACØ8
ACØ7
ACØ6
ACØ5
ACØ4
ACØ3
ACØ2
ACØ1
ACØØ