SYSTEMS CONCEPTS

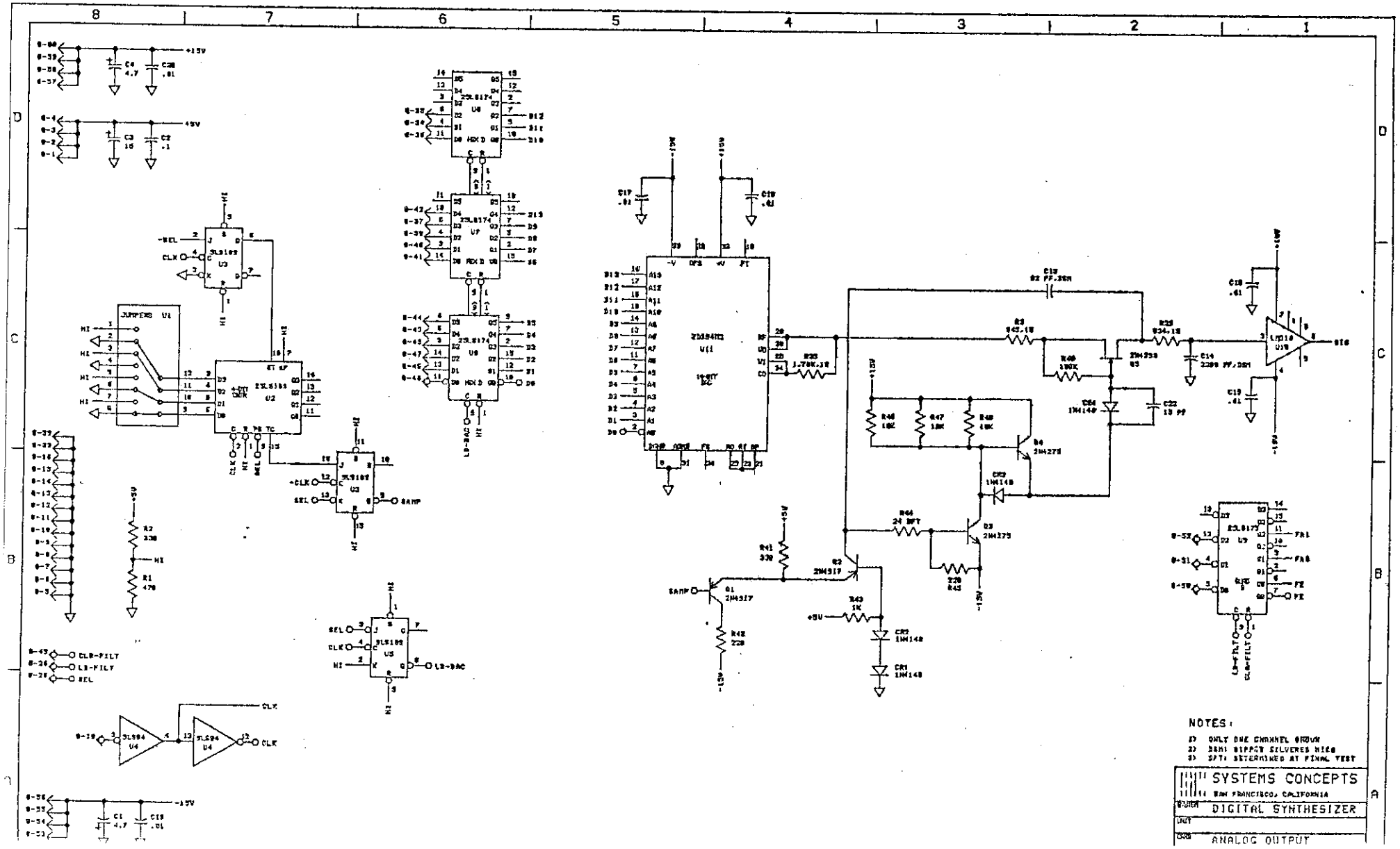
520 THIRD STREET SAN FRANCISCO, CALIFORNIA 94107

# DIGITAL SYNTHESIZER
# PRELIMINARY
# ENGINEERING DRAWINGS
# VOLUME 1

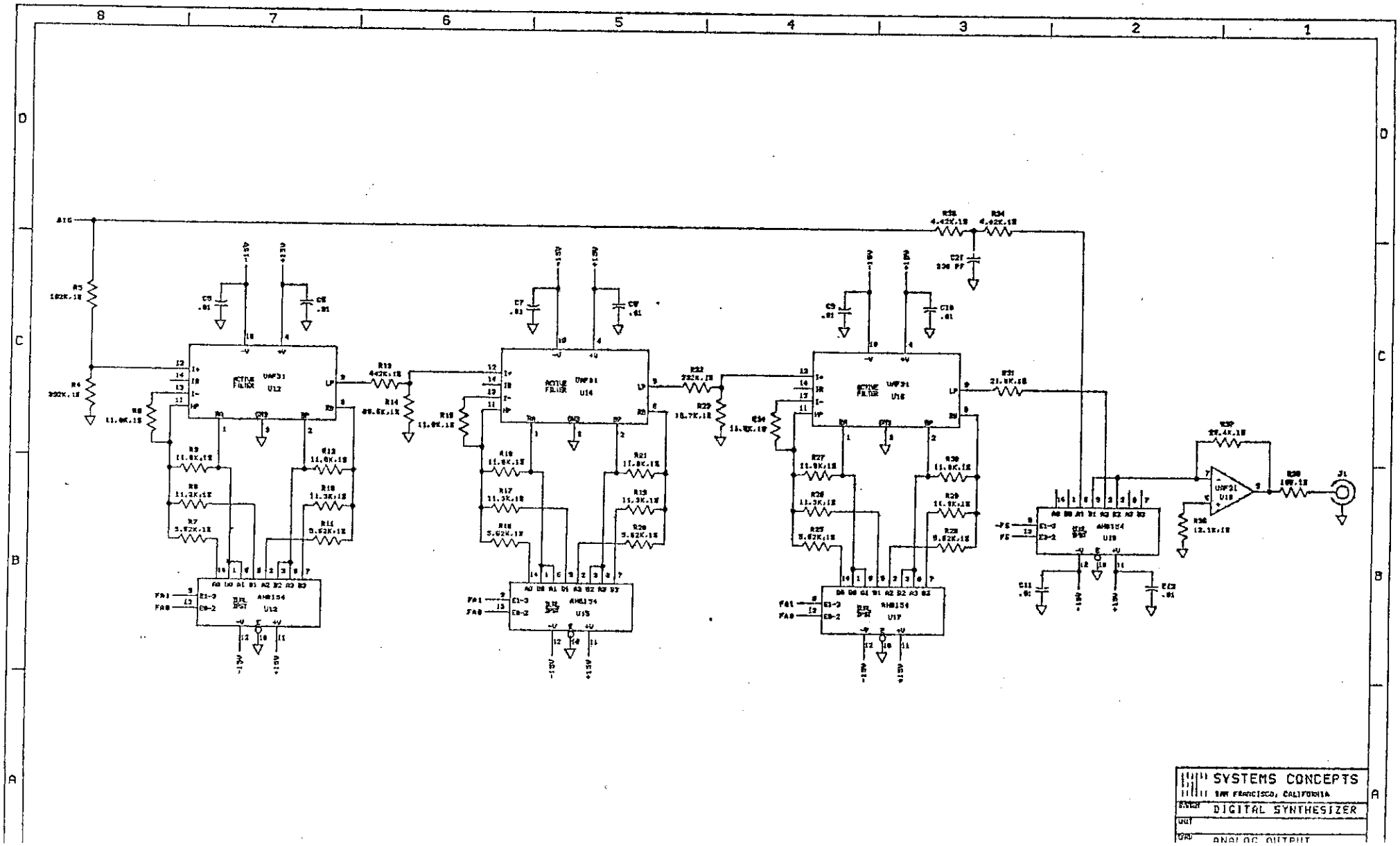


TELEPHONE: 415-442-1500 TWX: 910-372-6062

SFE; ANLOG1

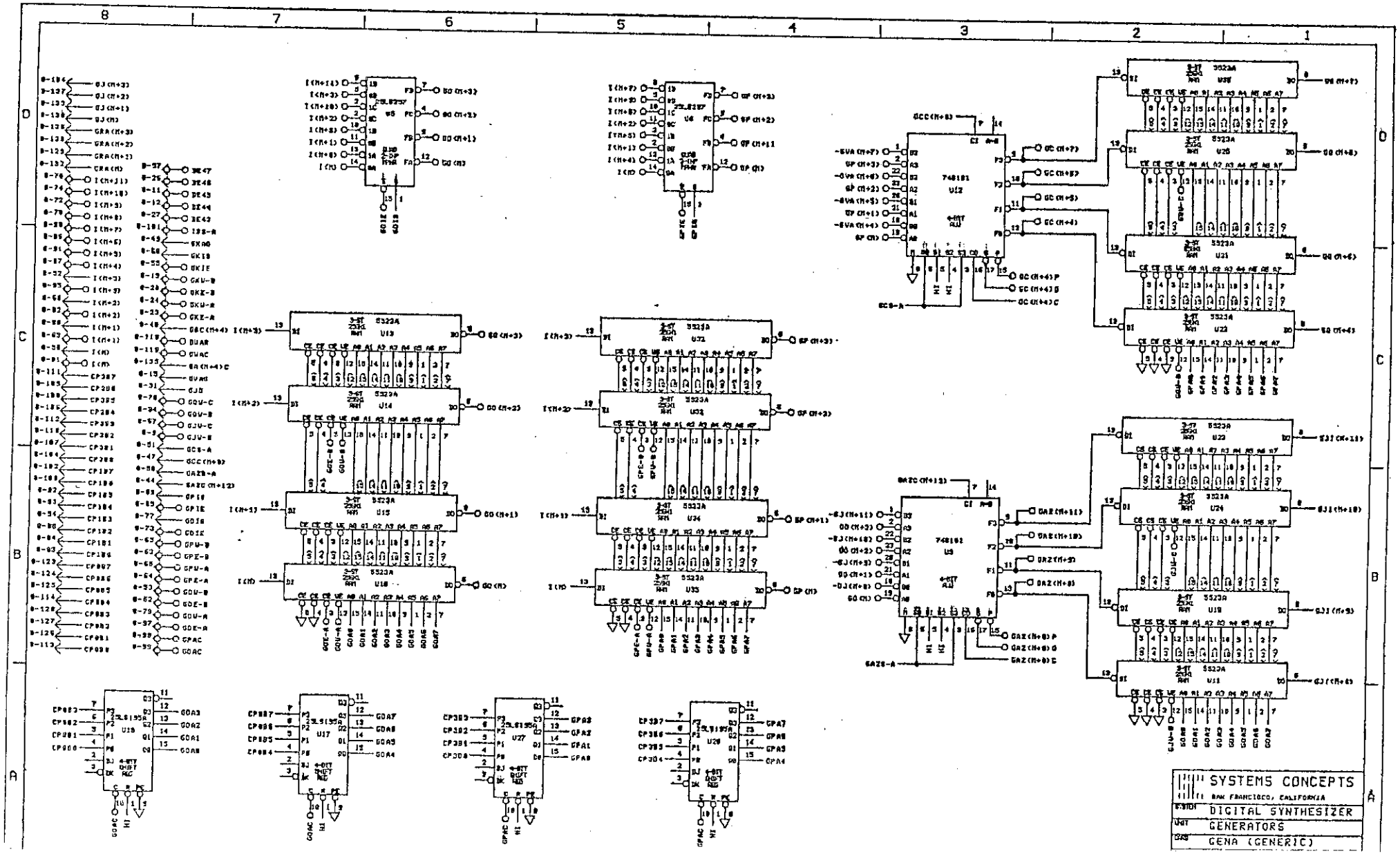

NOTES:
1) ONLY ONE CHANNEL SHOWN
2) DAN: DIFFER SILVERES HIGH
3) DFT: DETERMINED AT FINAL TEST
SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
ANALOG OUTPUT

SFE; ANLOG 2


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
ANALOG OUTPUT

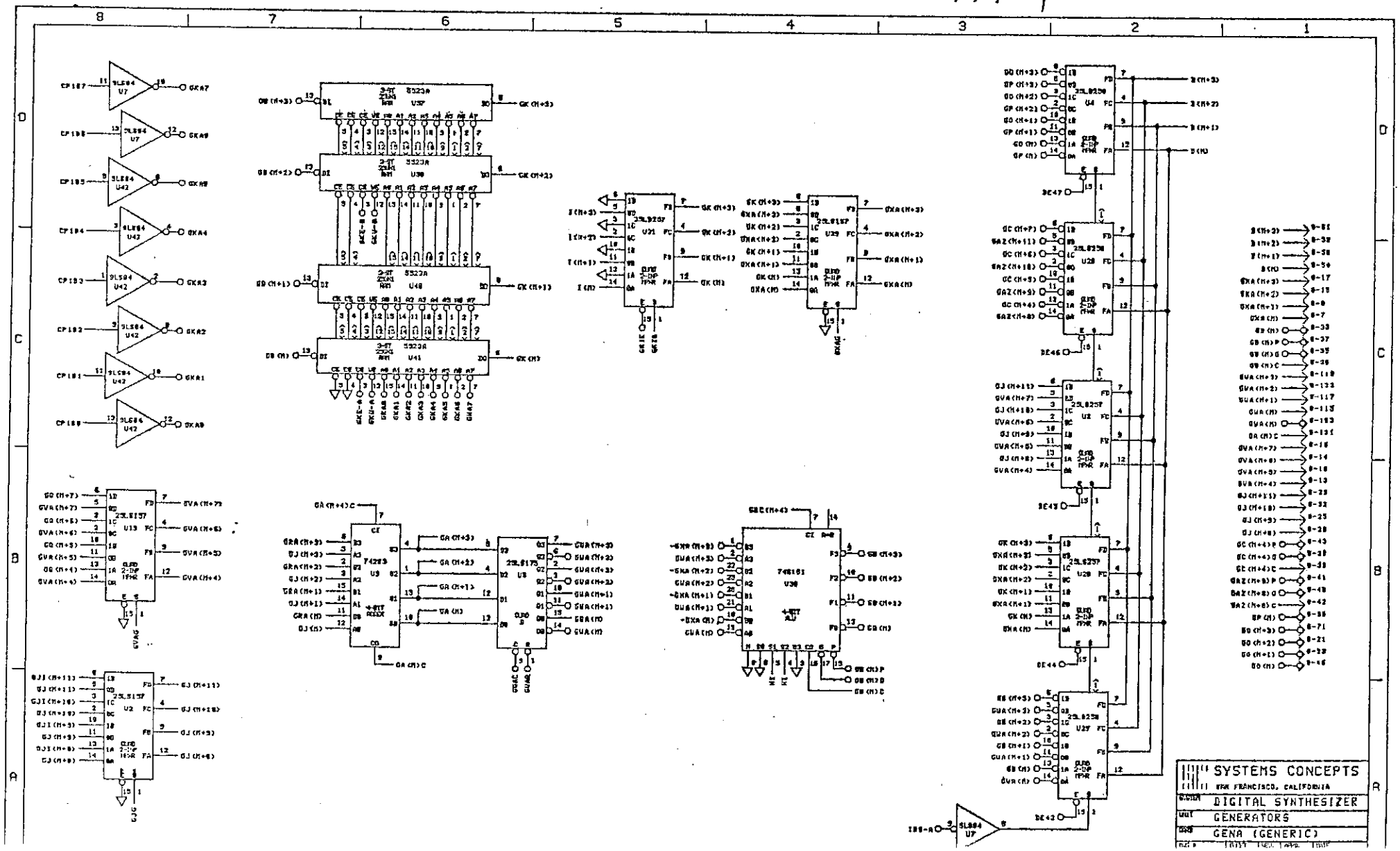
SWB; GENA1

PTF=1



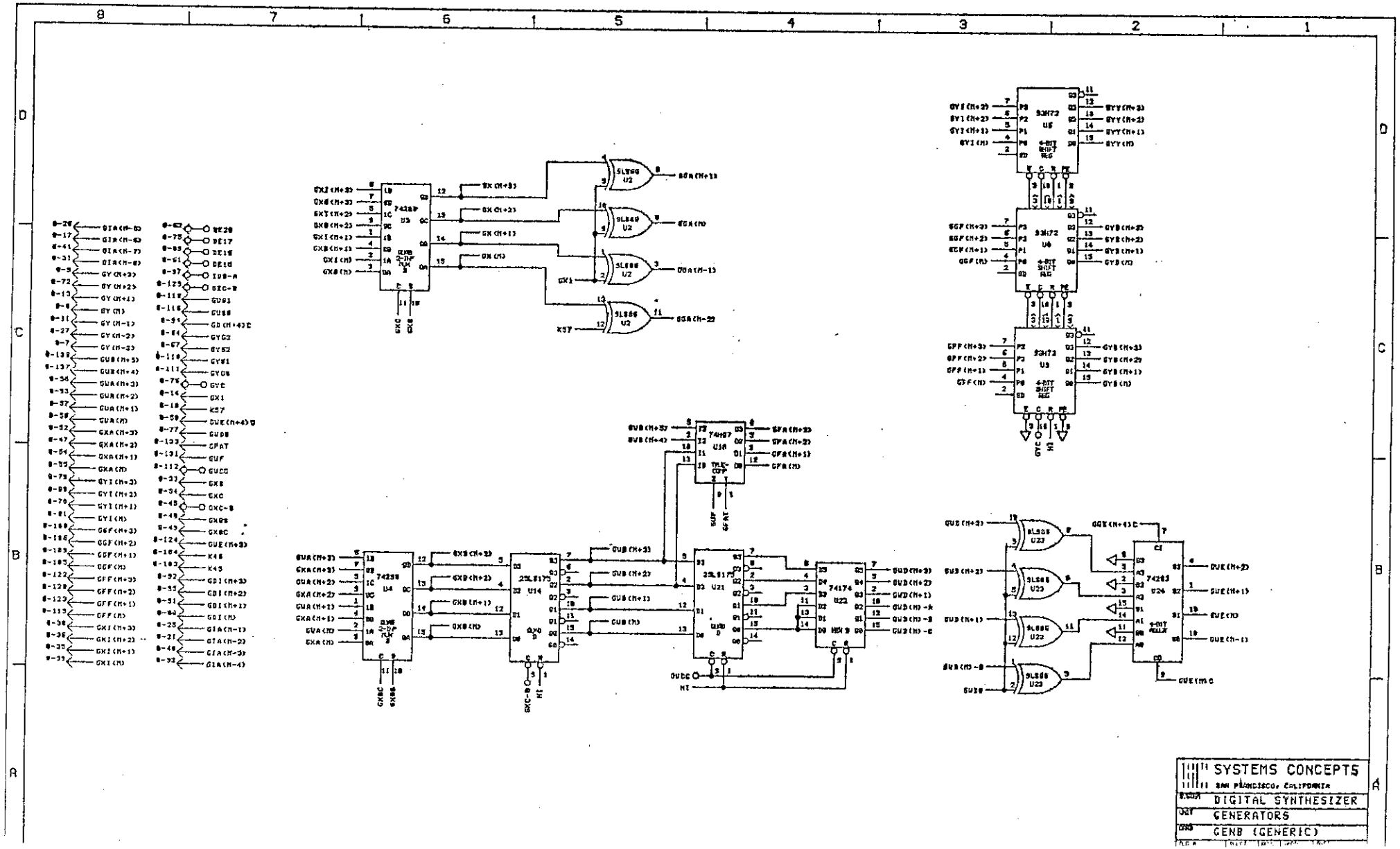
SW B; G ENA 2

PTF=1



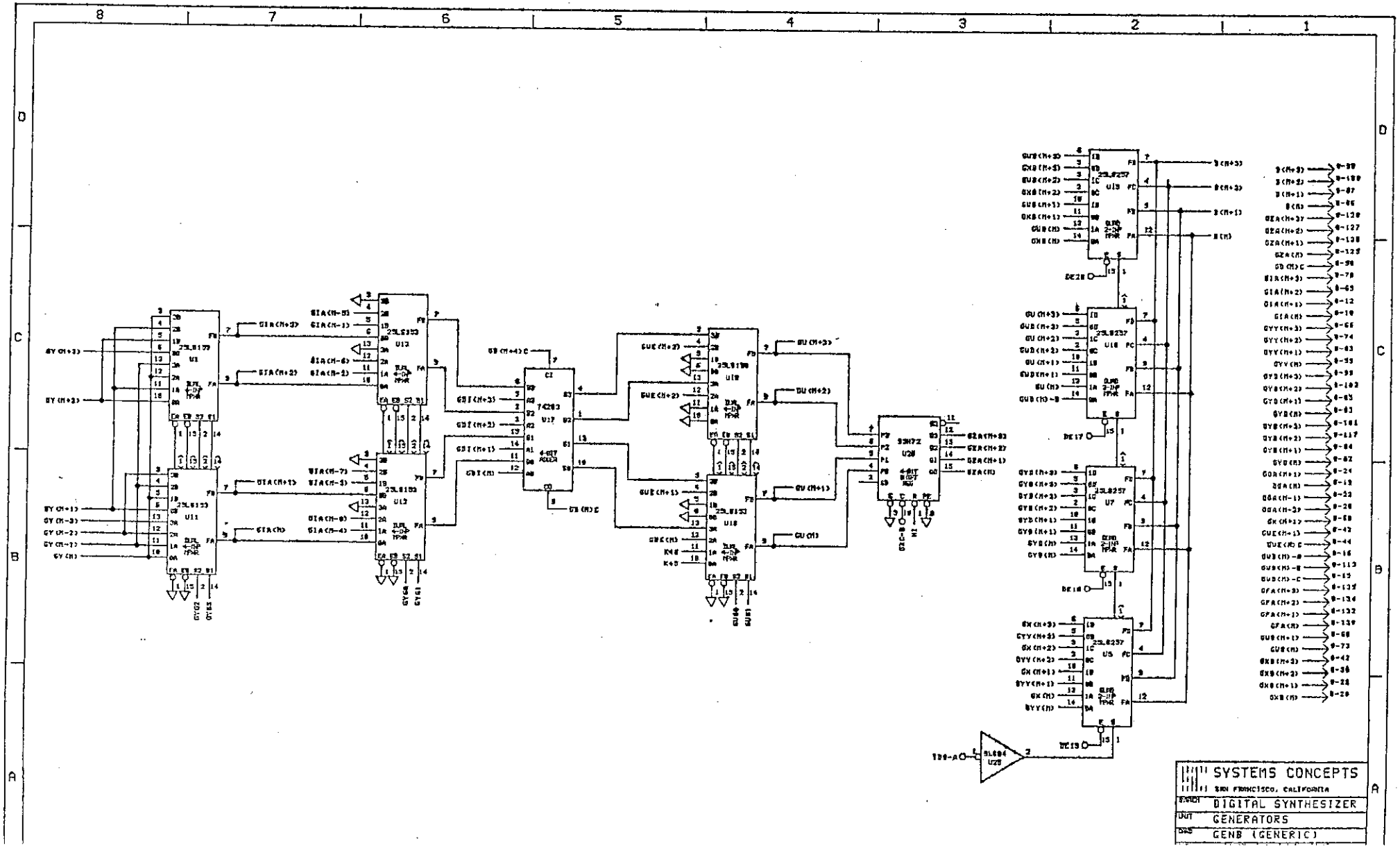
SWB; GENB1

PTP=1



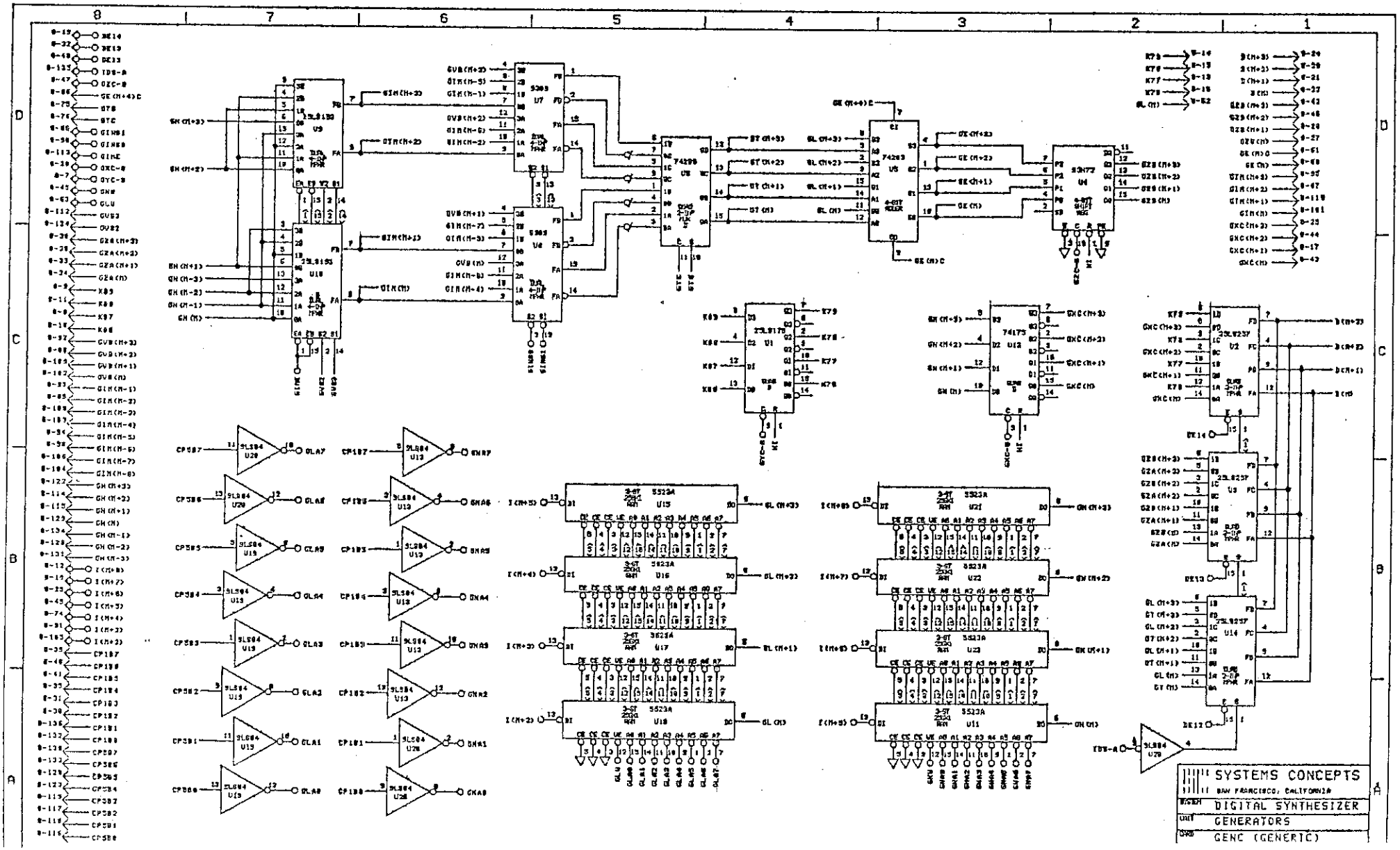
SNB; GENB2

PTF=2

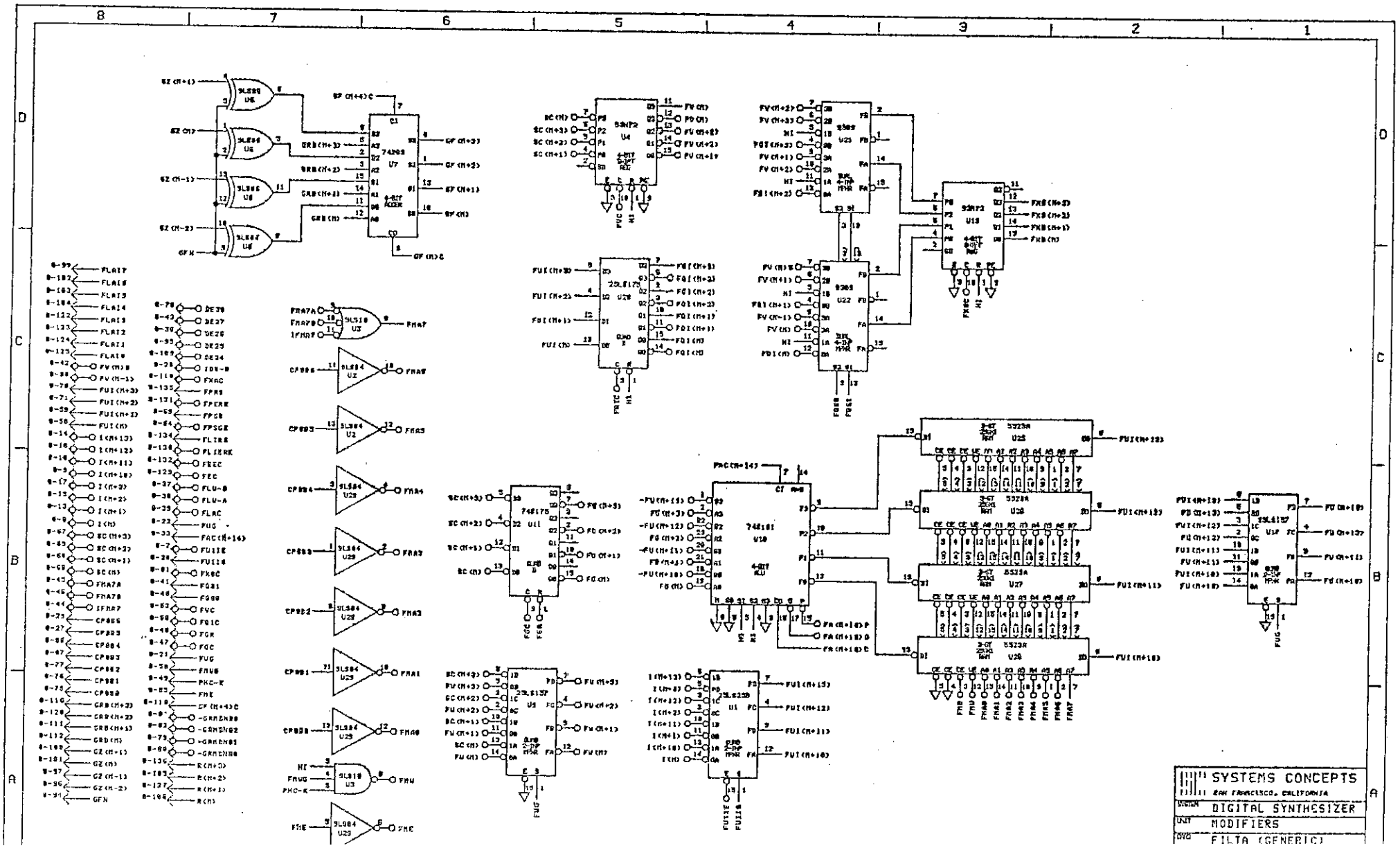


SFE; GENC

PTF=1

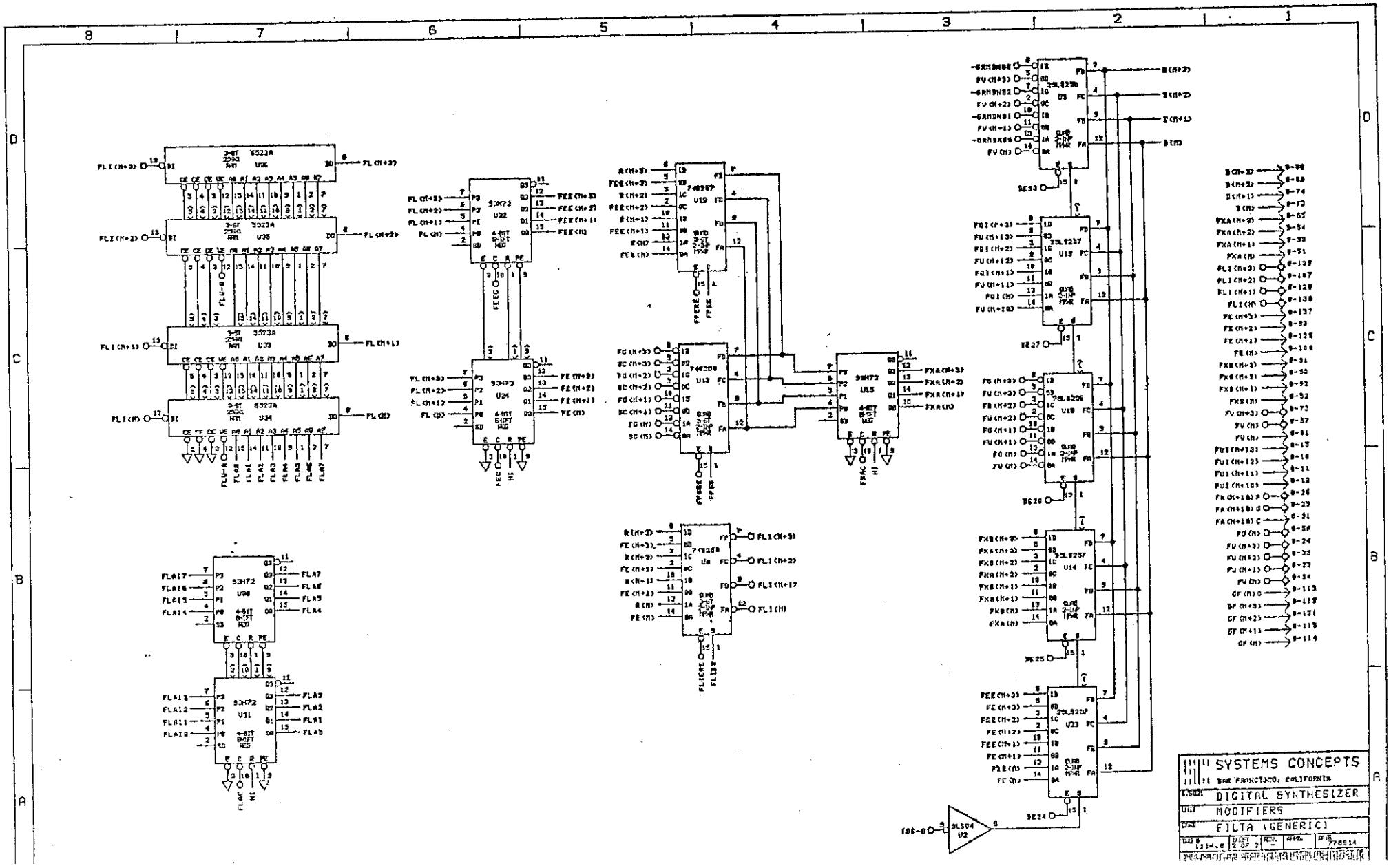


SFE; FILTA1
PTF=1

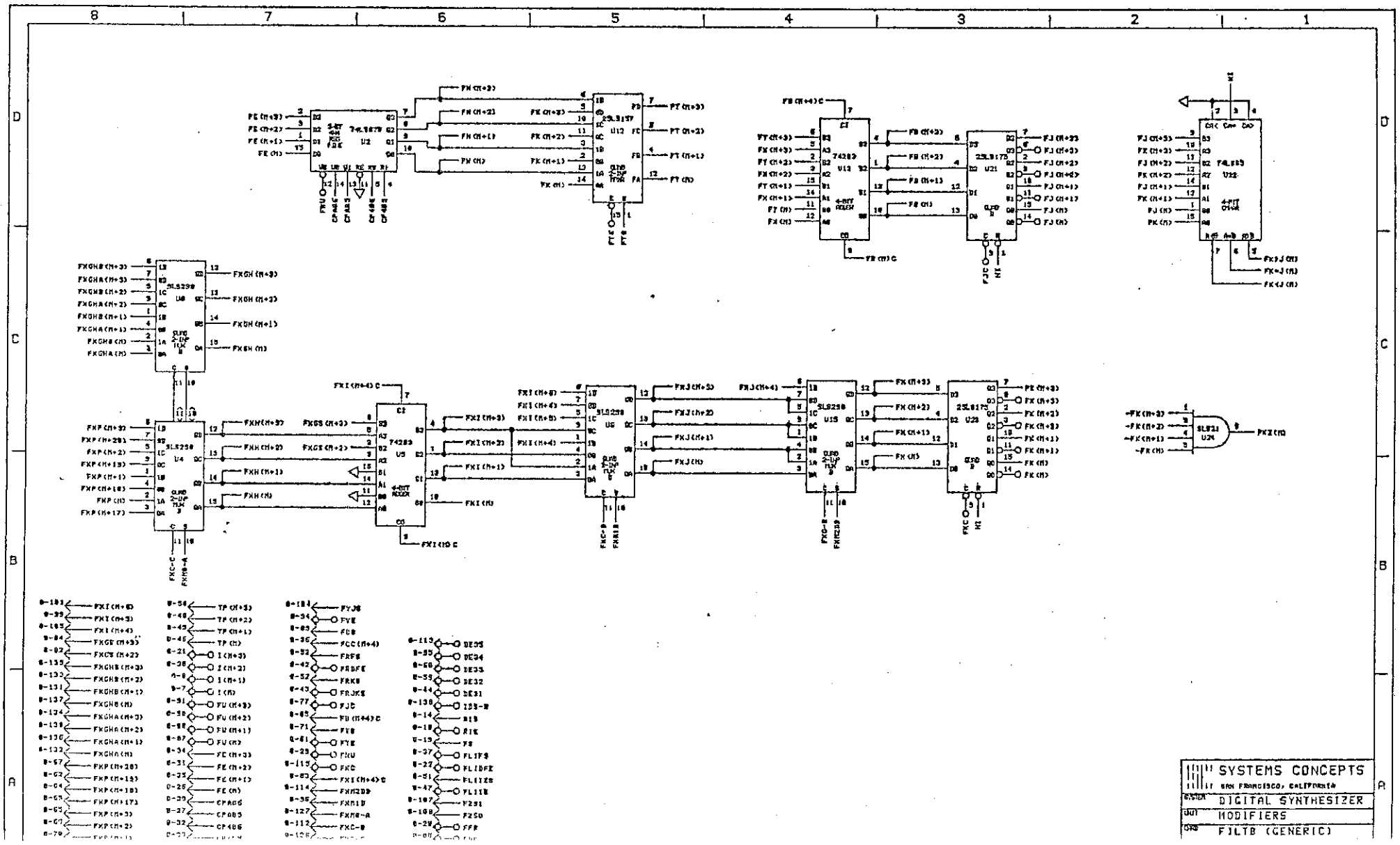


SFE; FILTA2

PTF=1



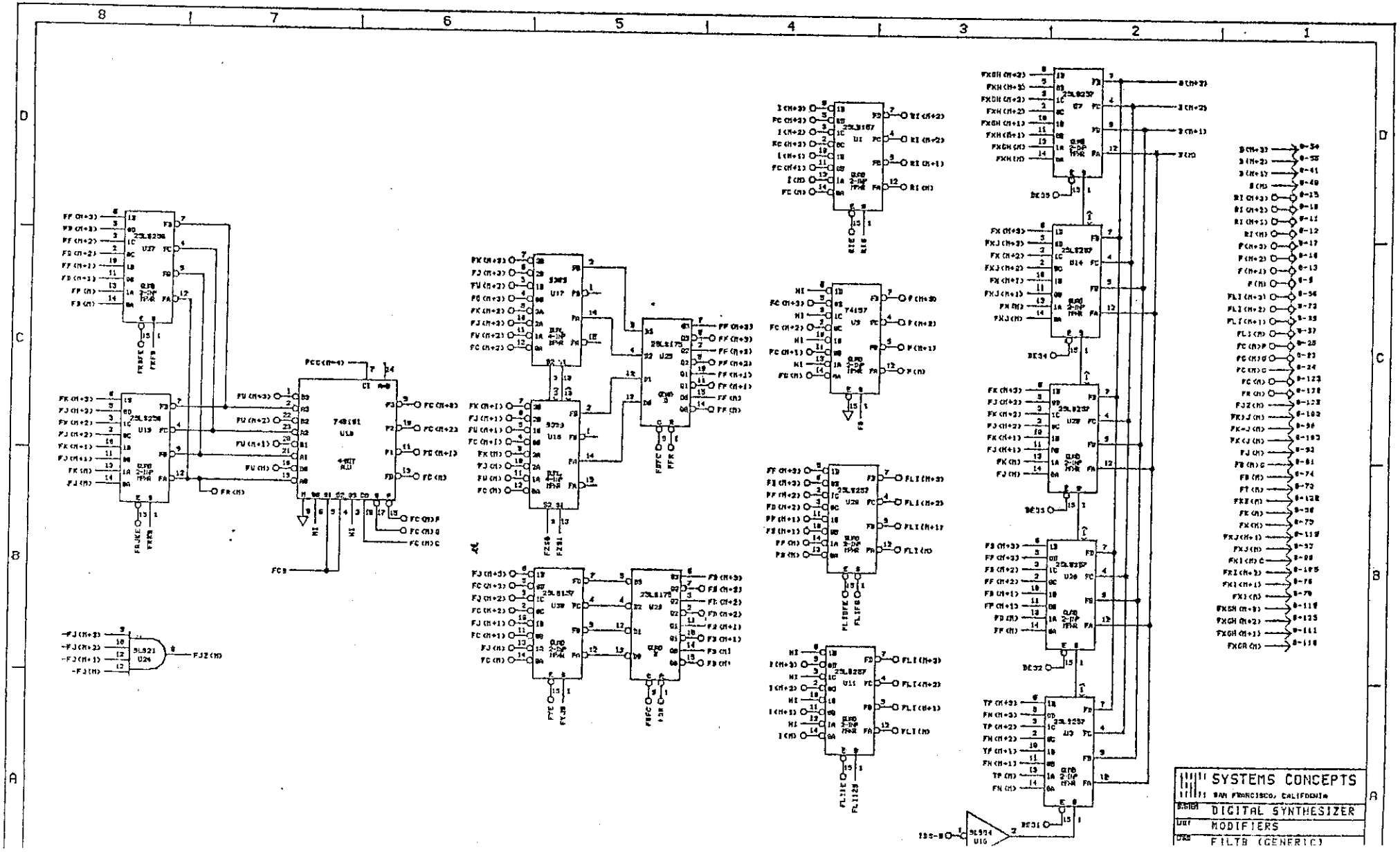
SFE), FILTB1
PTF=1



SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
MODIFIERS
FILTB (GENERIC)

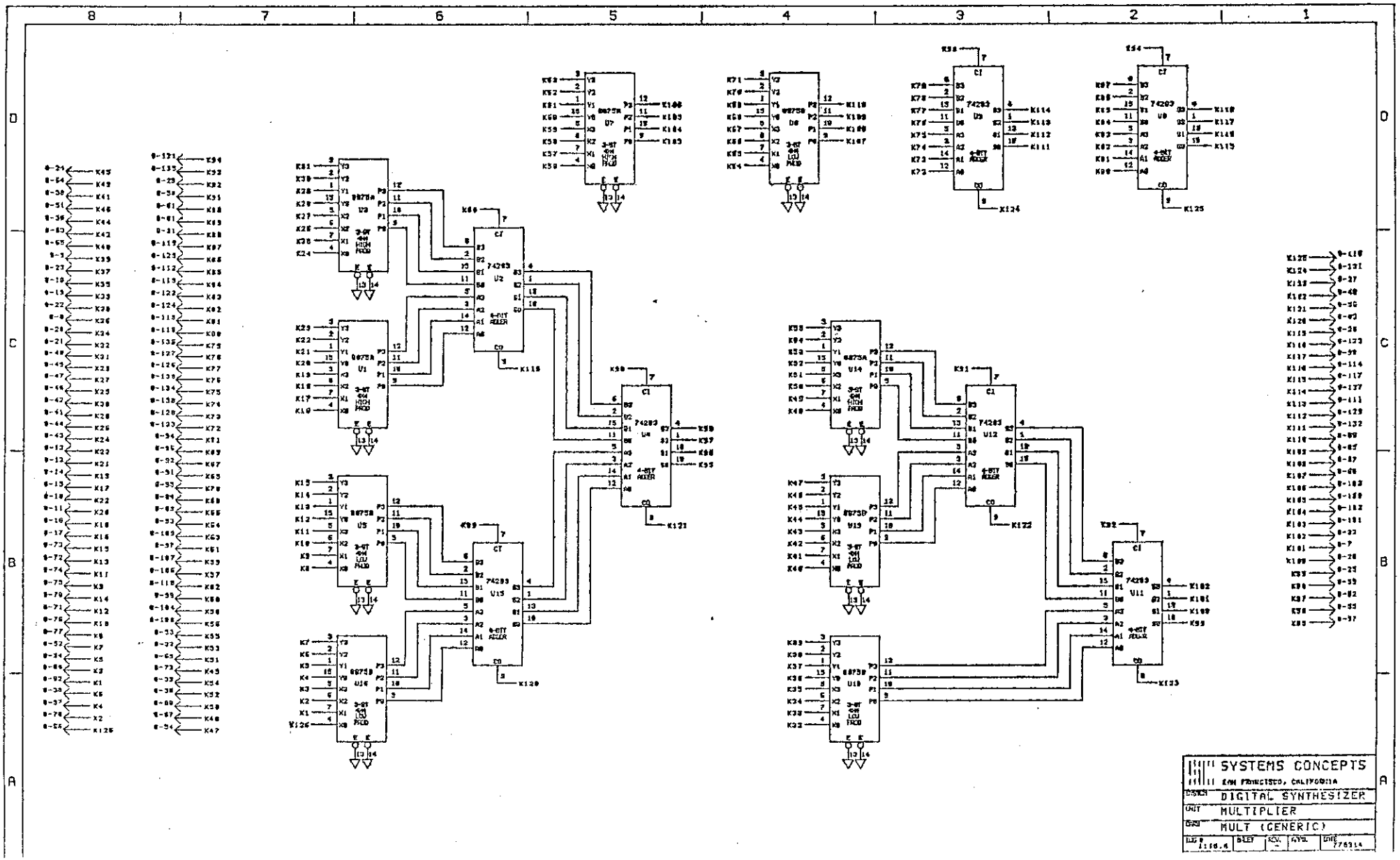
SFE; FILTB2

PTF=1



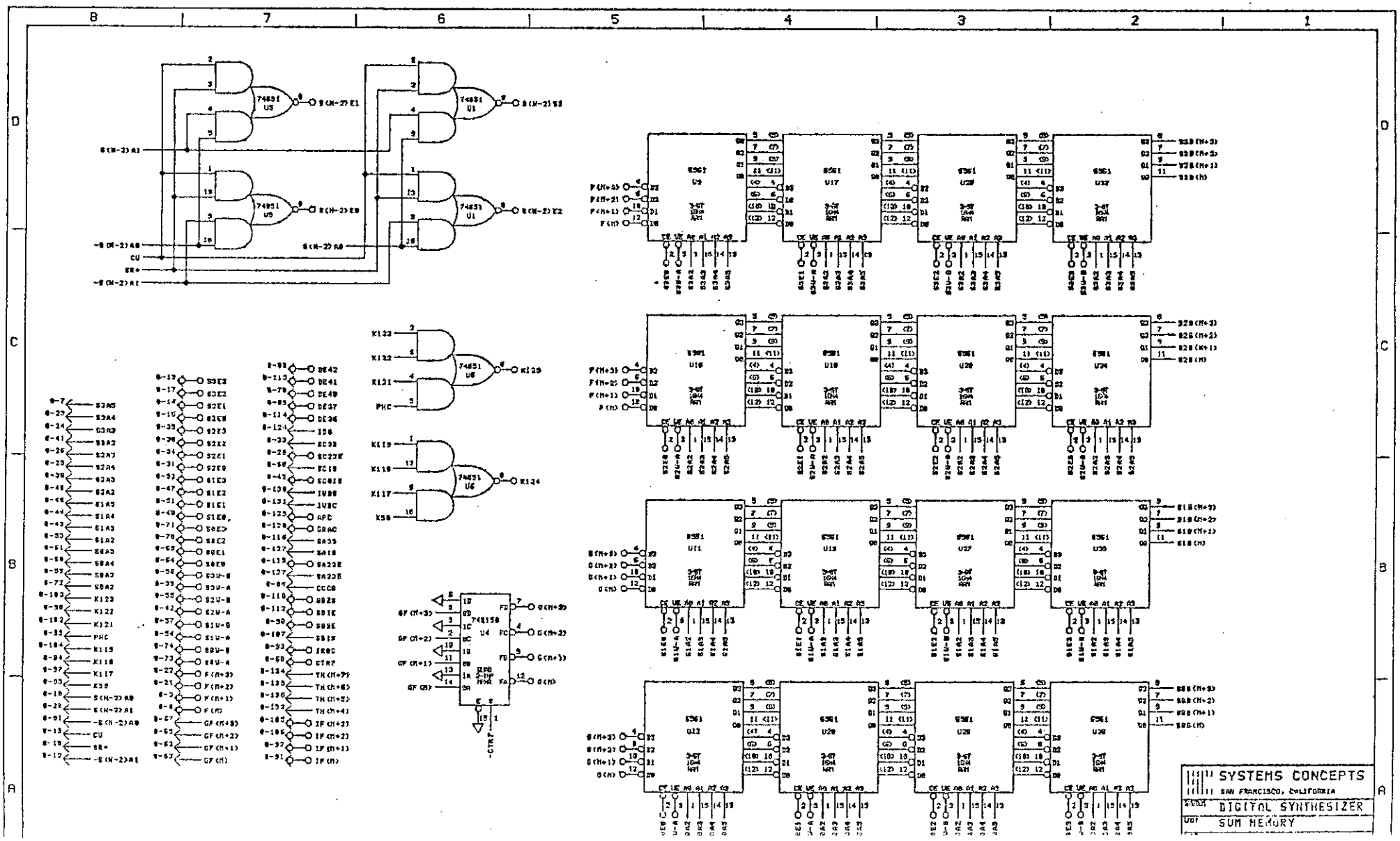
DM; MULT

PTF=1


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
MULTIPLIER
MULT (GENERIC)

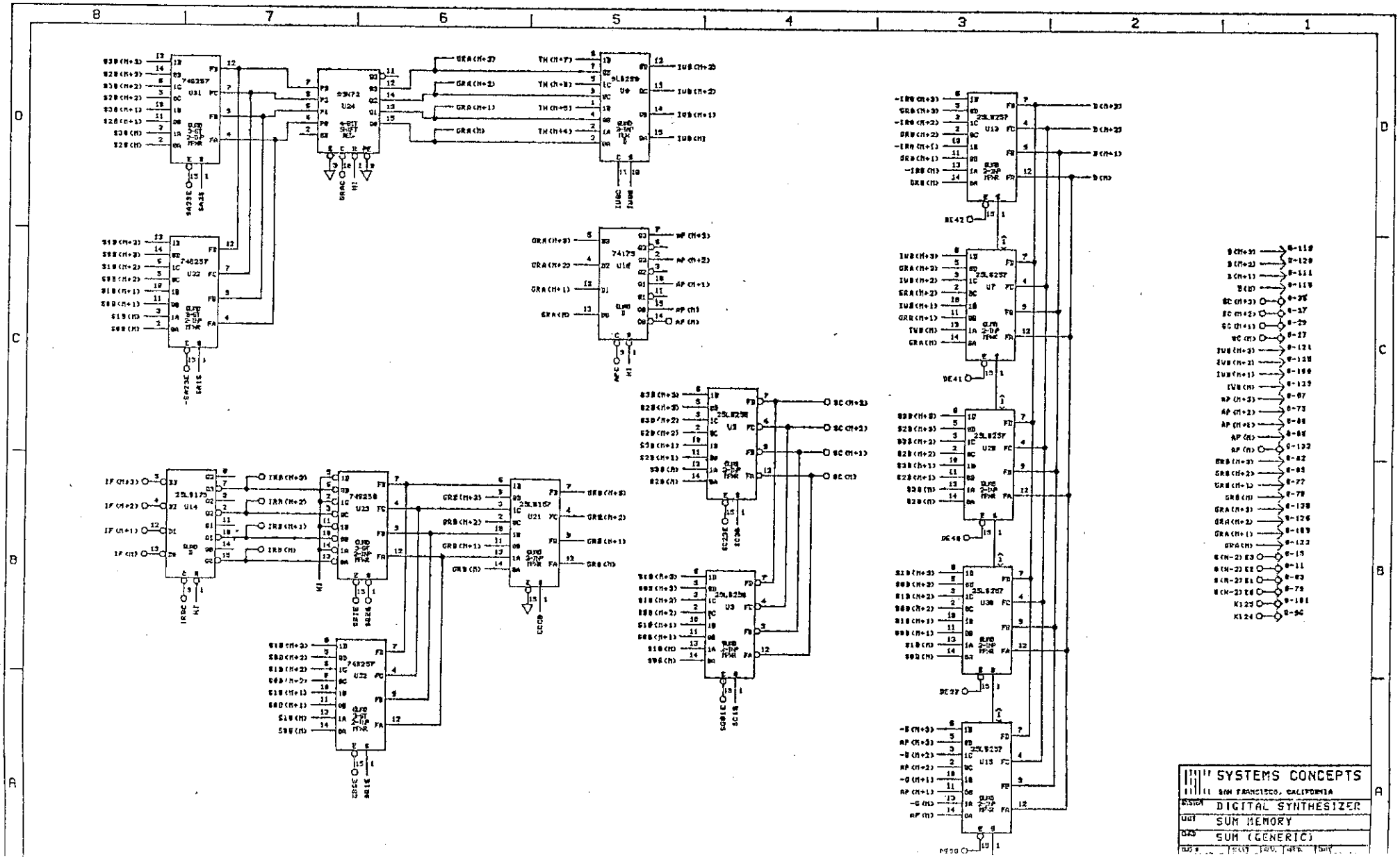
SFE; SUM 1

PTF=1



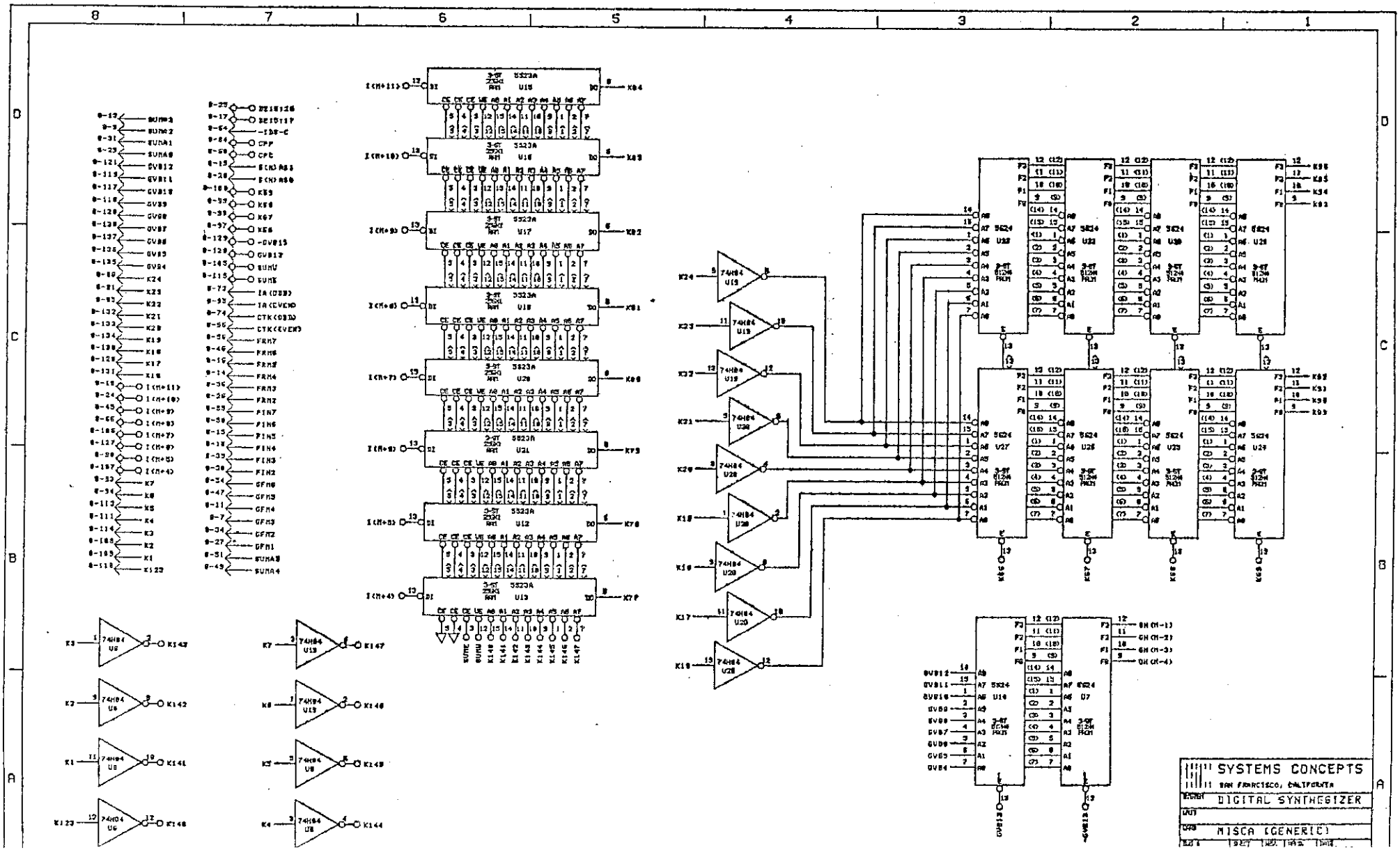
SFE;SUM2

PTF=1



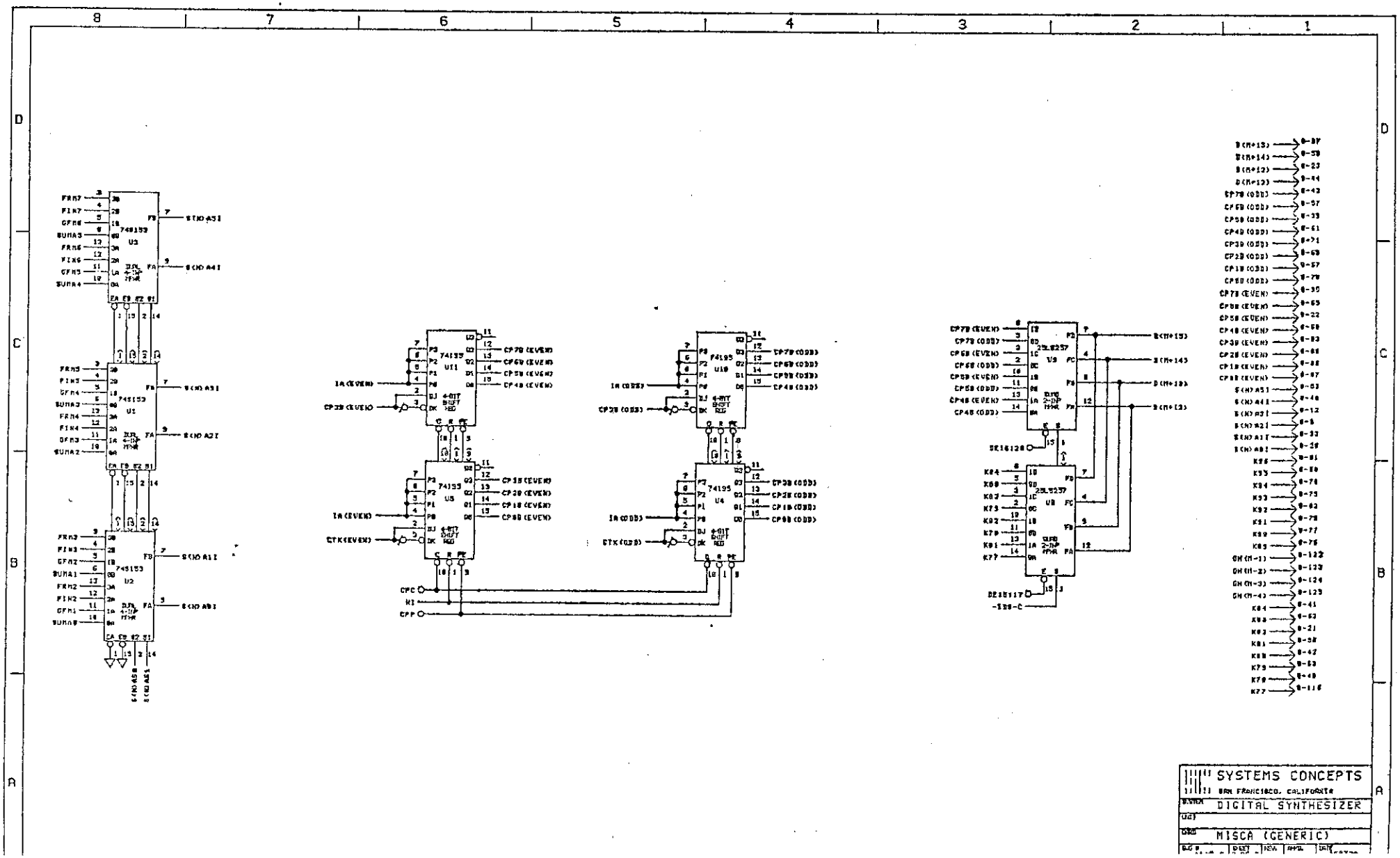
SRE; MISCA1

PTF=1



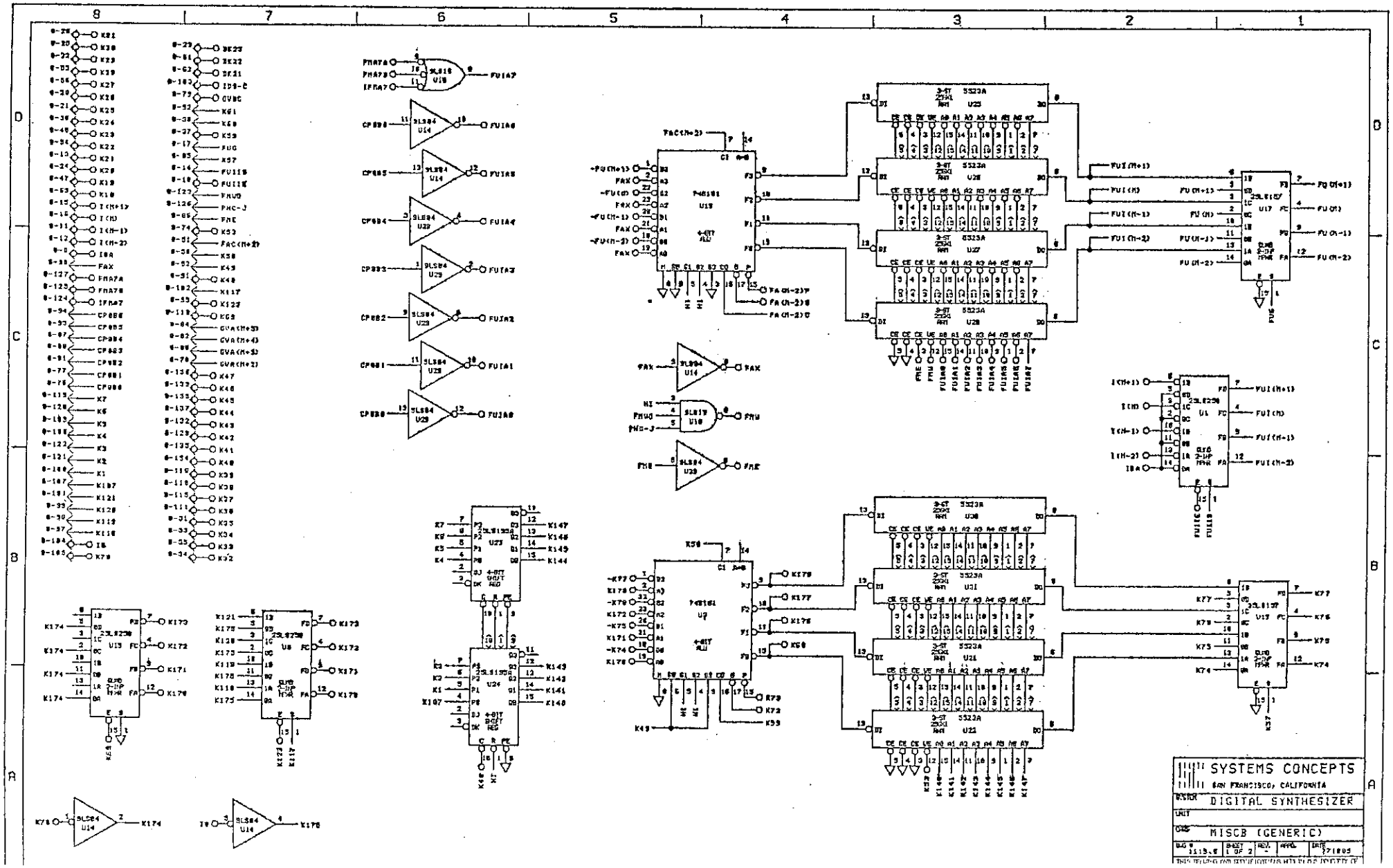
SRE; MISCA2

PTF=1



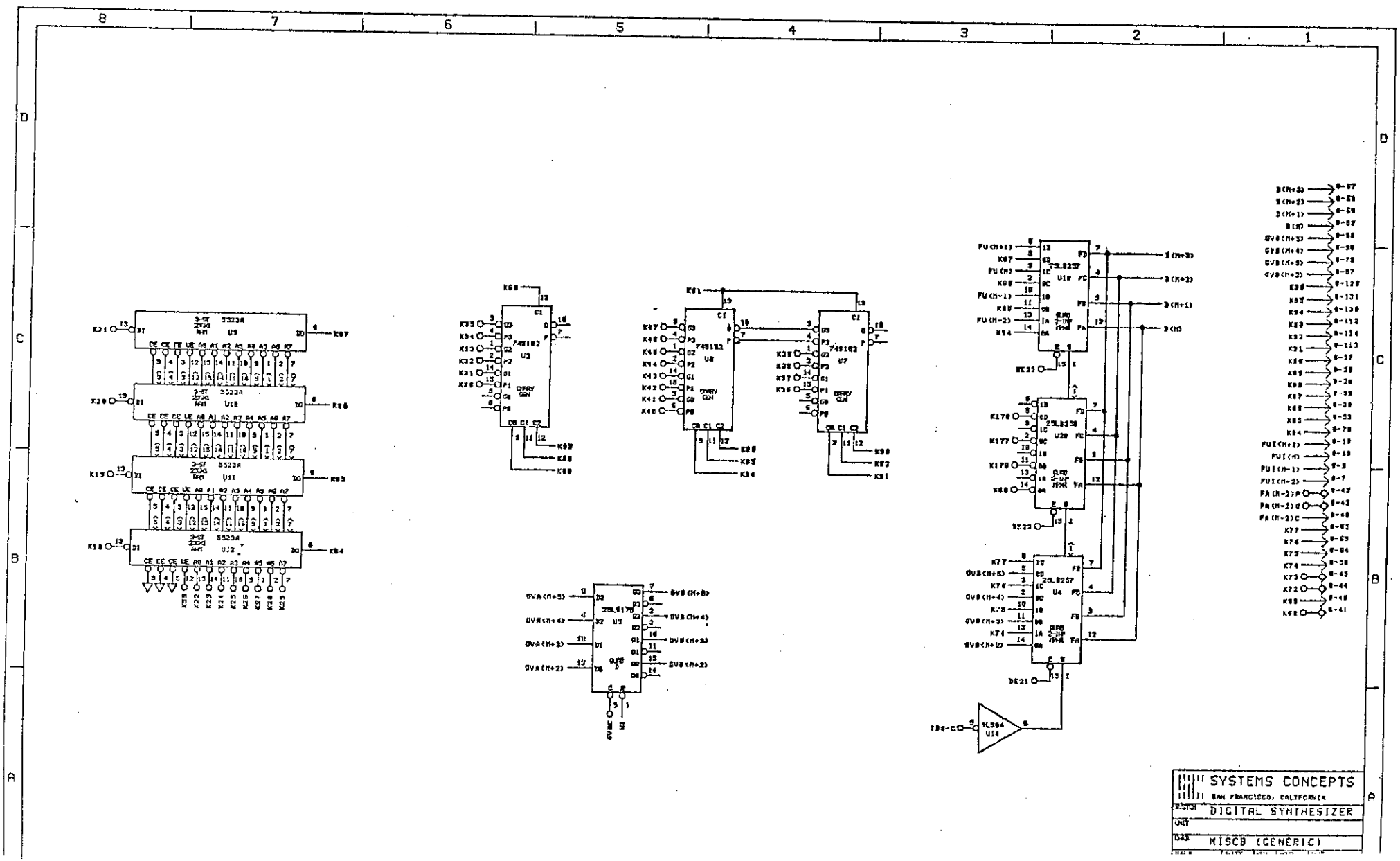
BG; MI 5CB1

PTF=1

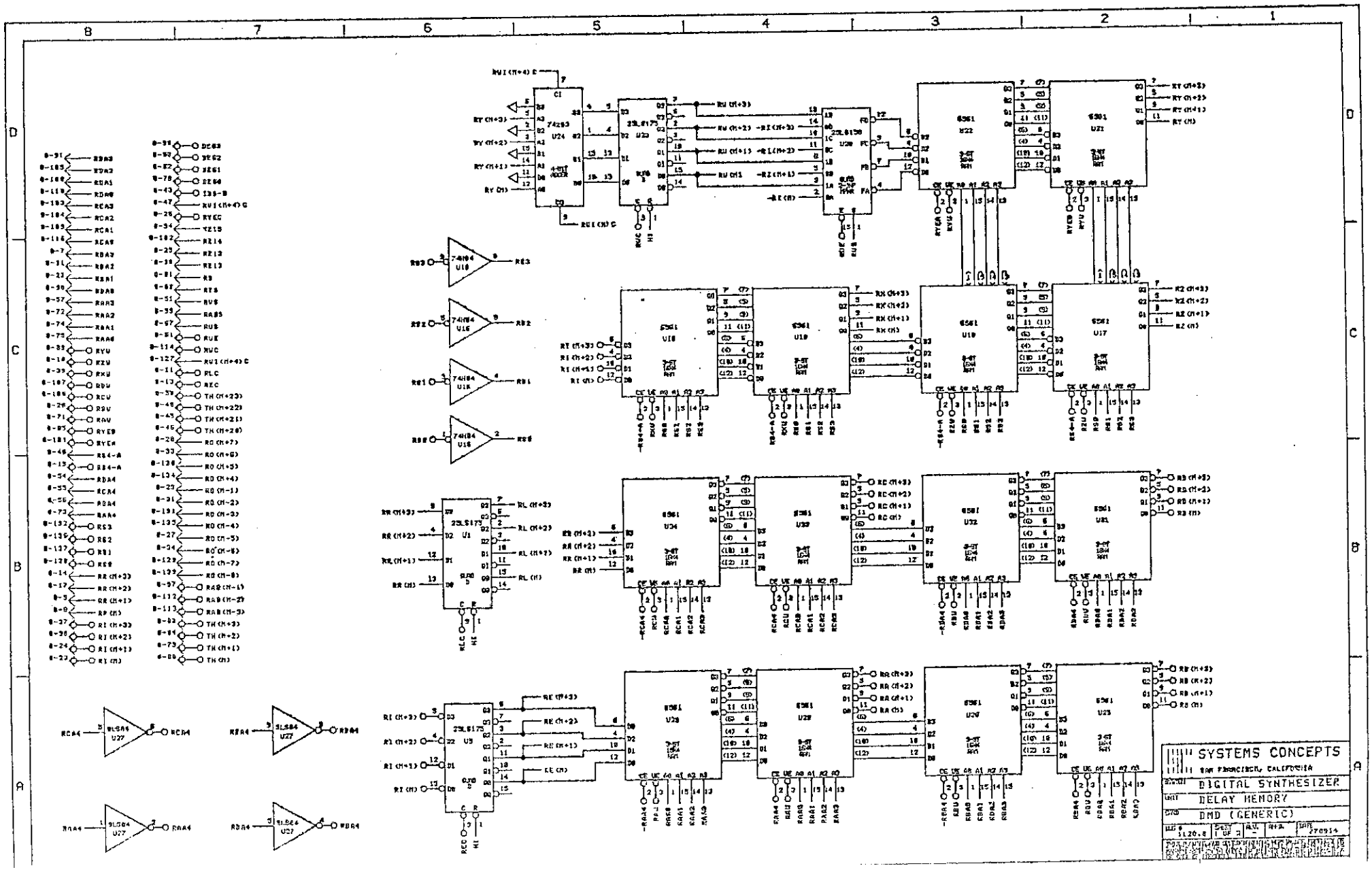


BG; MIS CB2

PTF=1

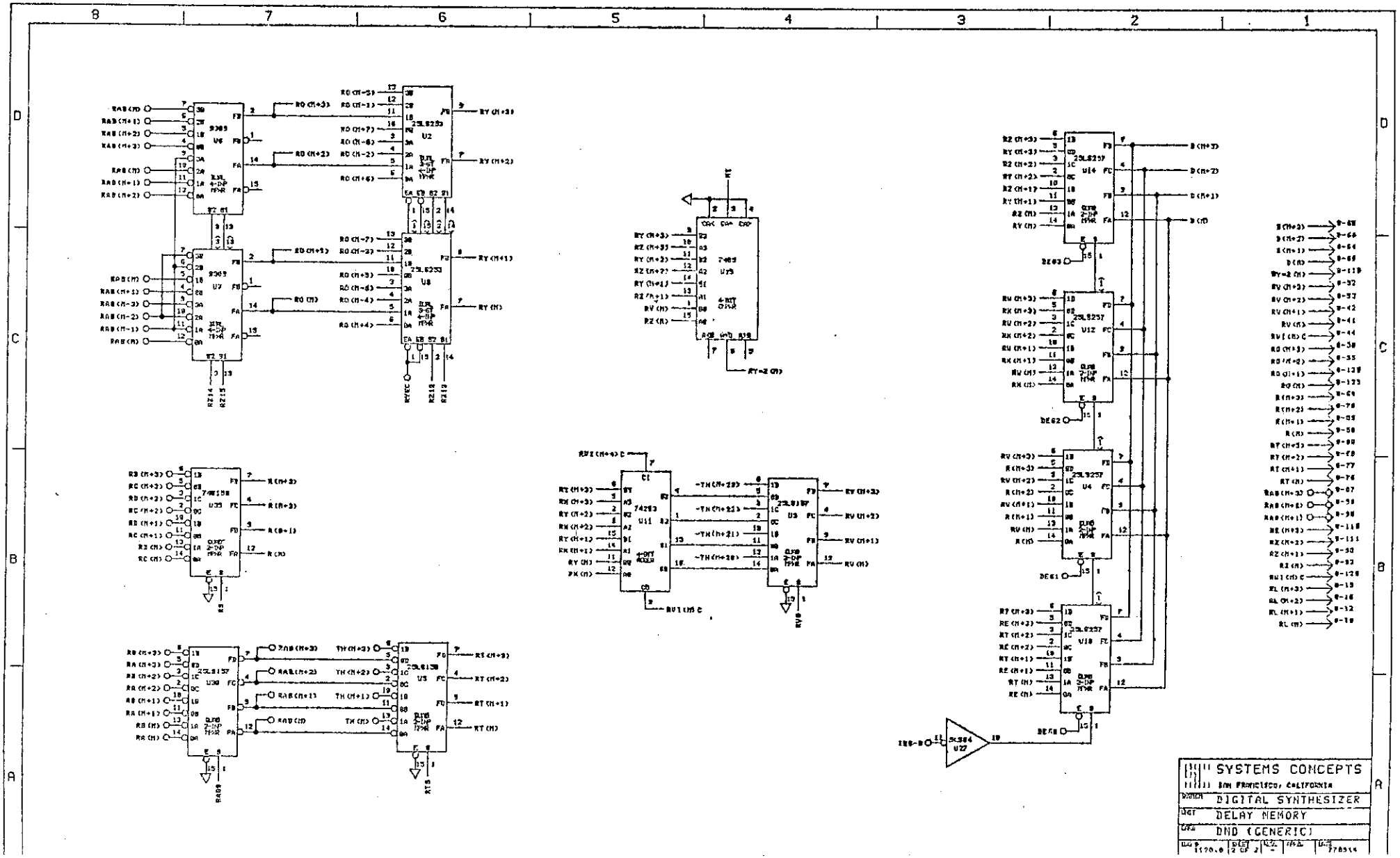


SWB: DMD1
PTF=1



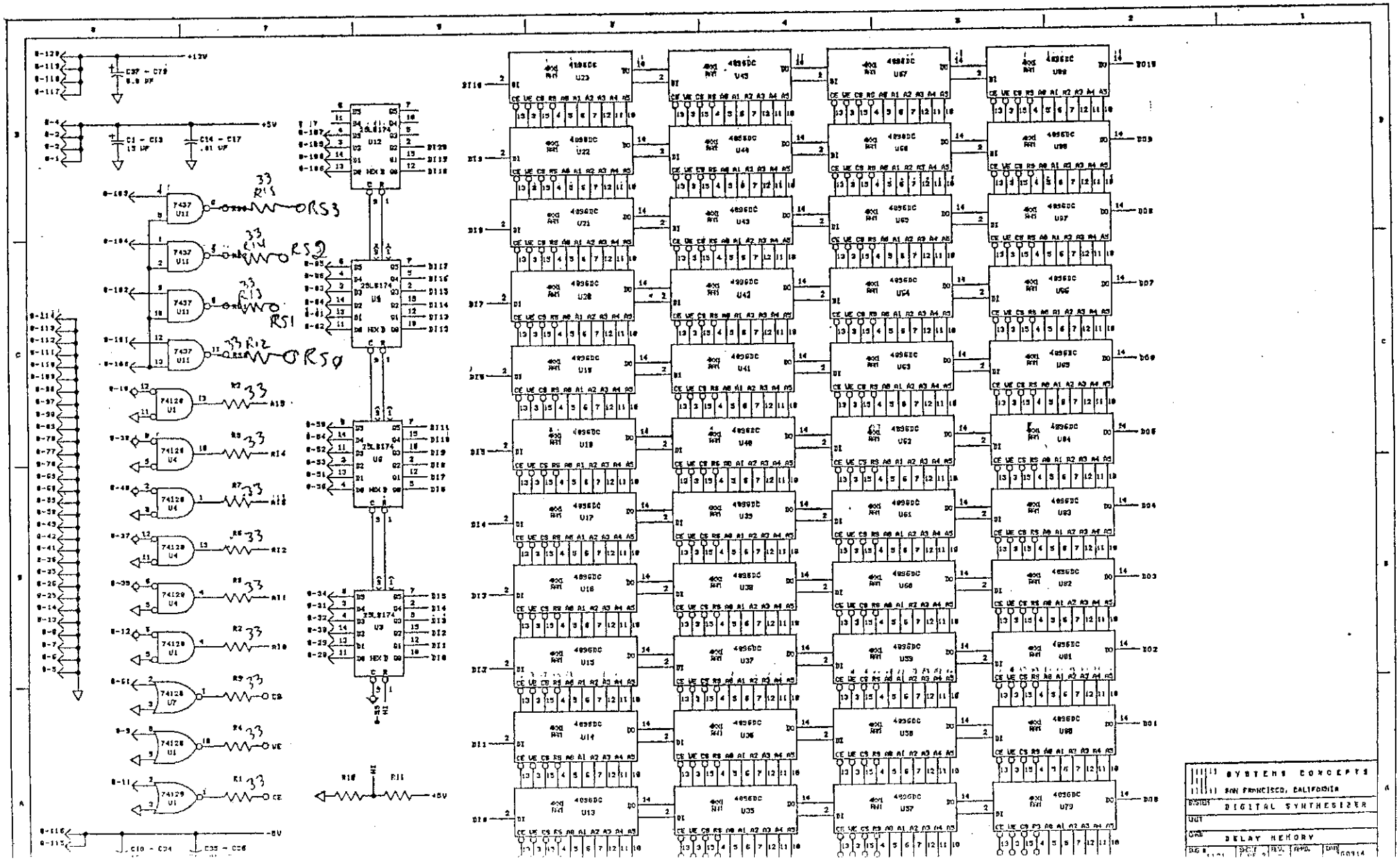
SWB; DMD 2

PTF = 1



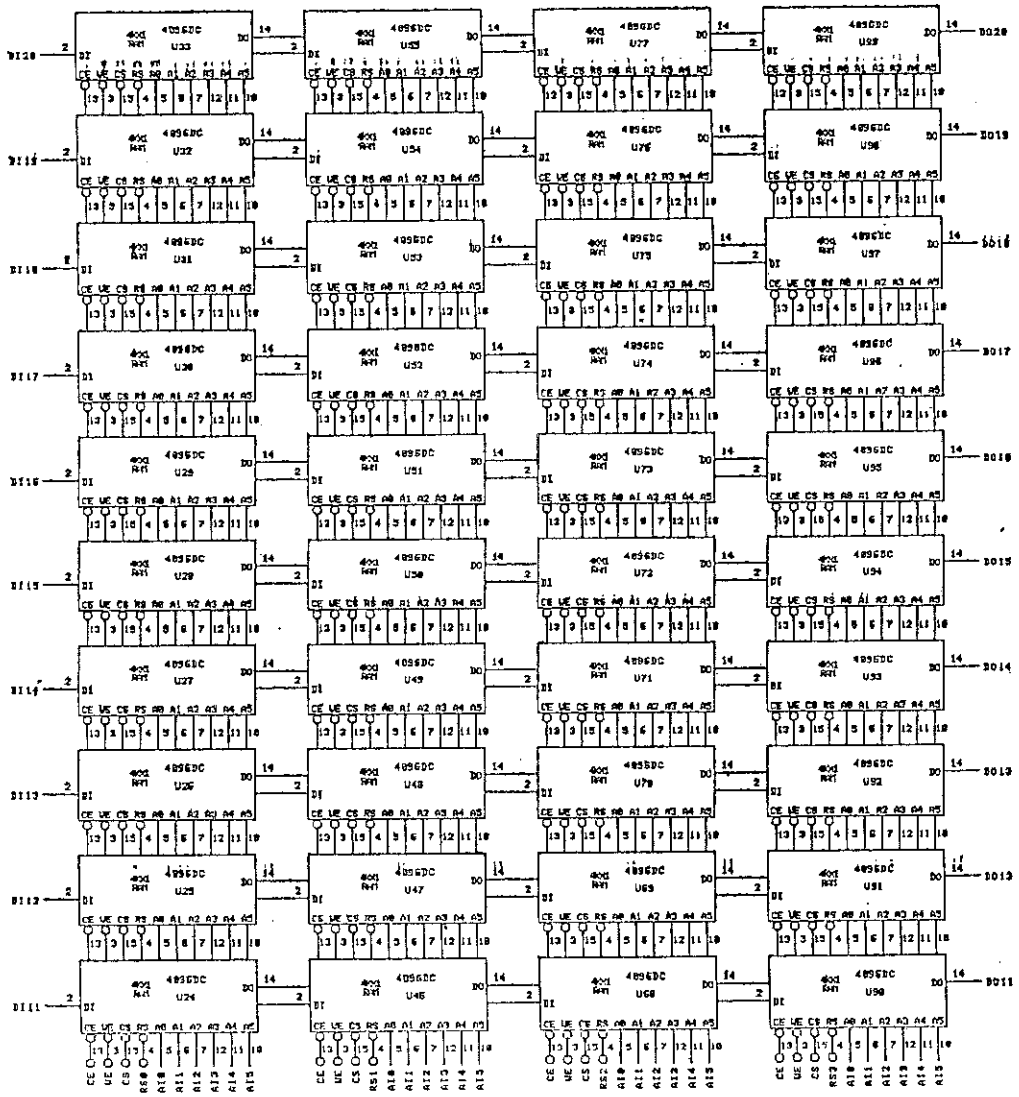
DXM; DEMEM 1

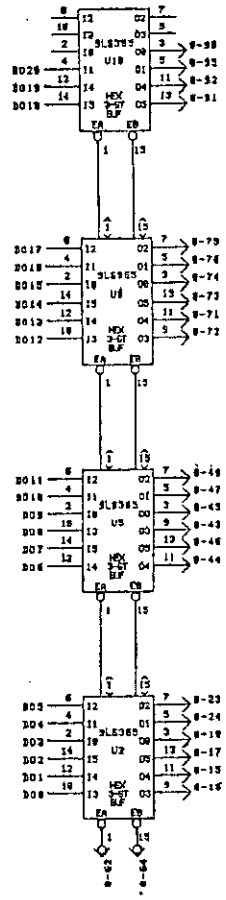
BIT 21 = Ø



DXM; DE MEM 2

BIT 21 = Ø



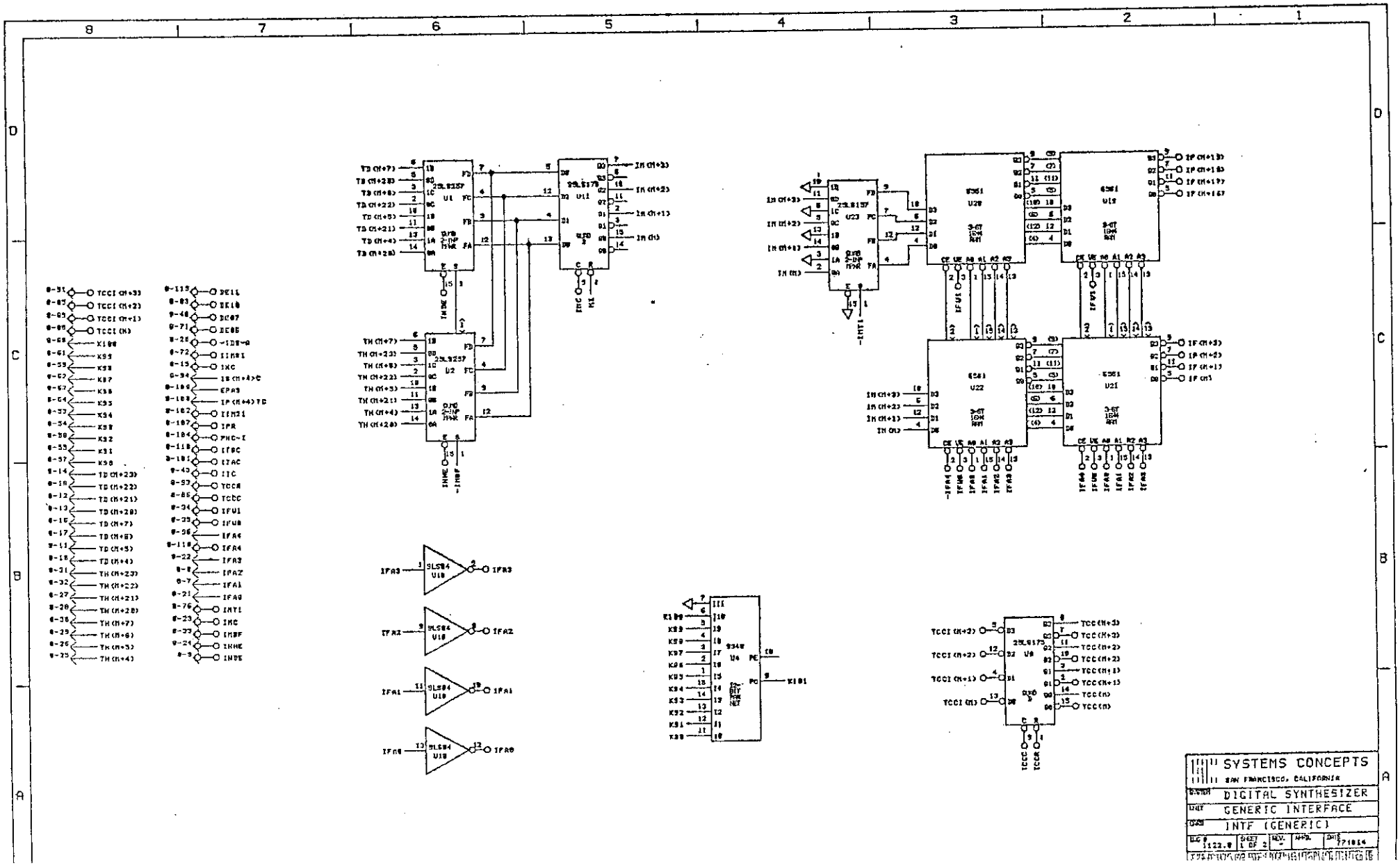

SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

DIGITAL SYNTHESIZER

DELAY MEMORY

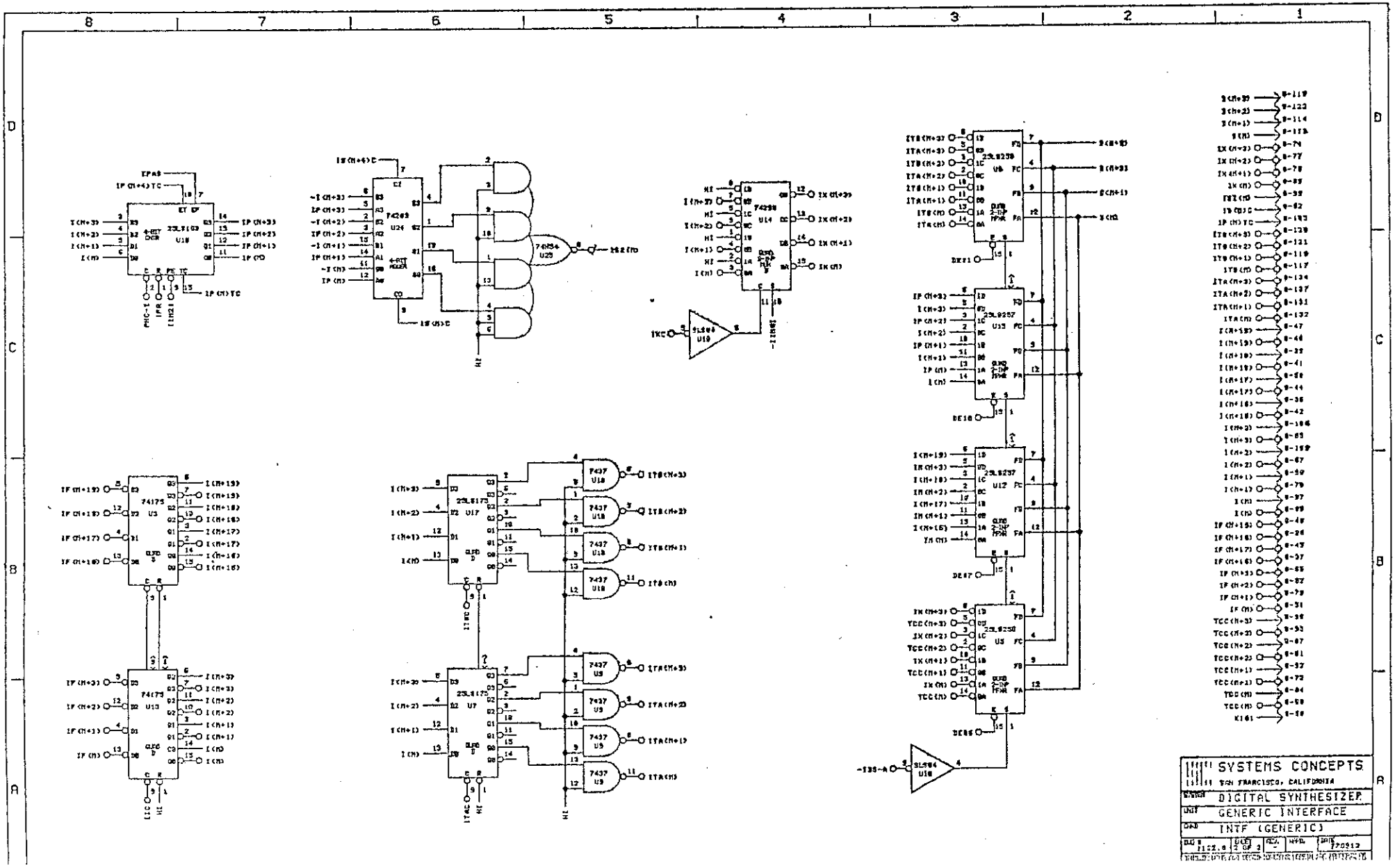
1121 | SHEET 2 OF 2 | REV. - | 700514

DM; INTF1
PTF=1

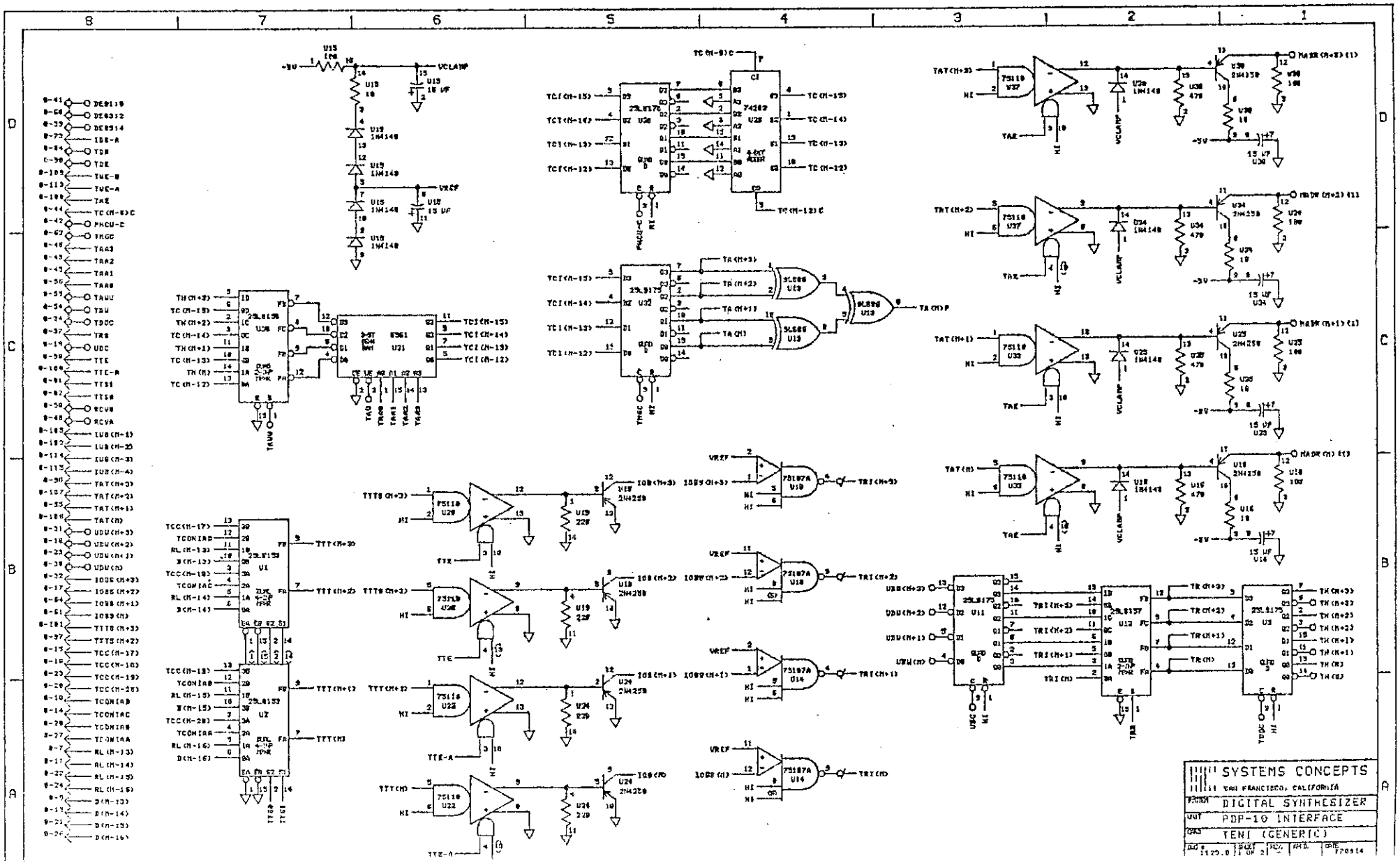


DM; INTF2

PTF=1

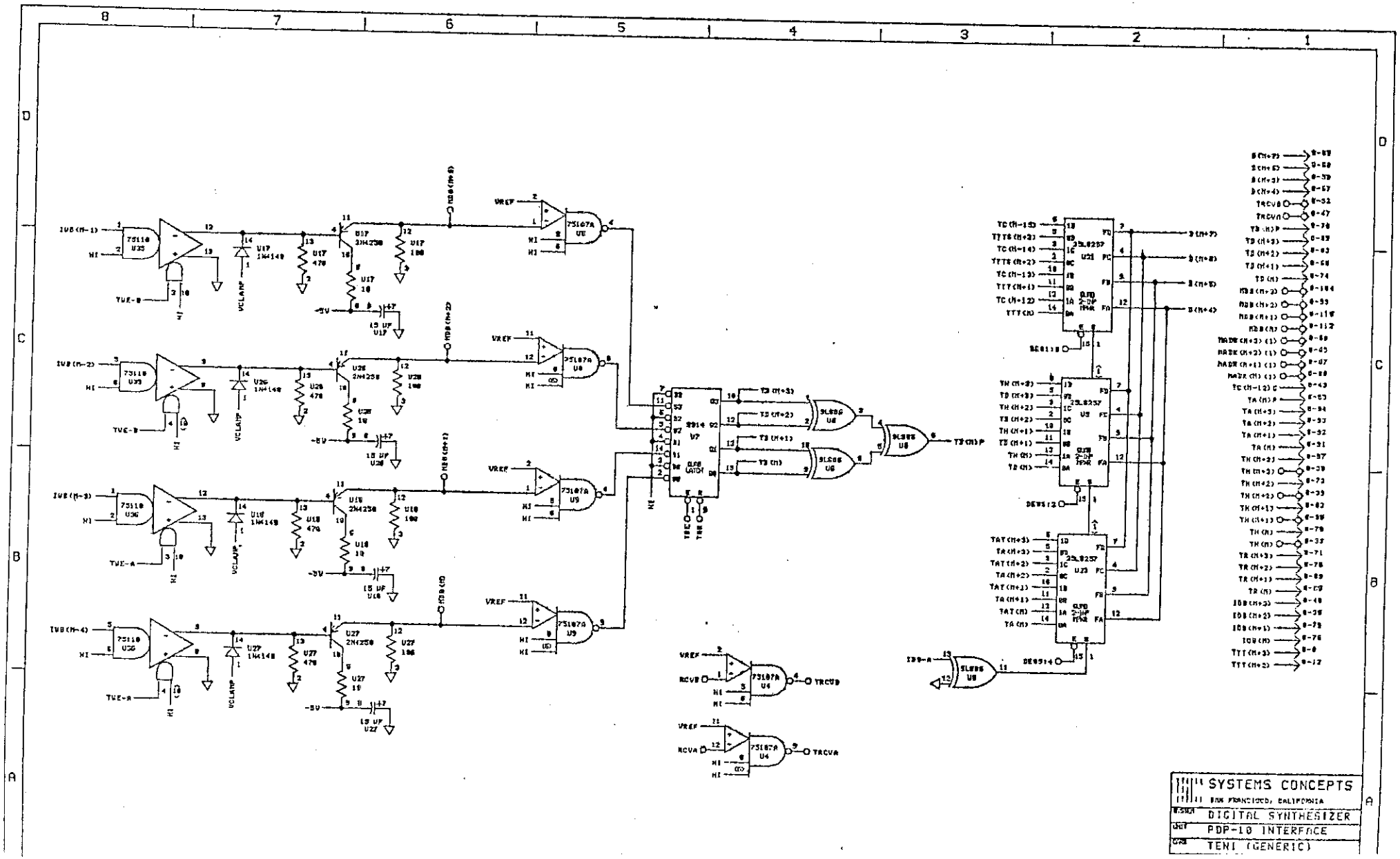


SWB; TENI
PTE=1

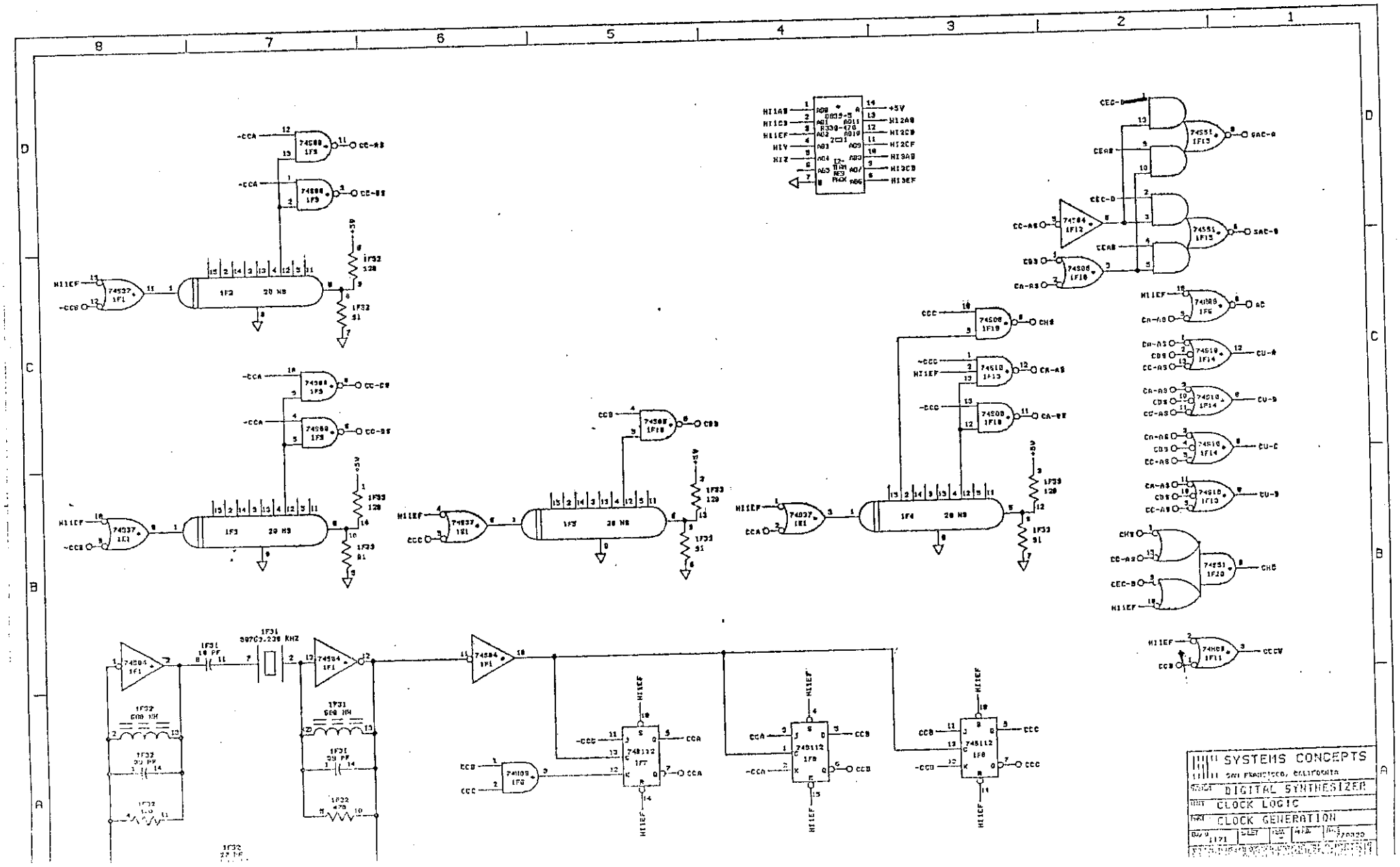


SWB, TENIL

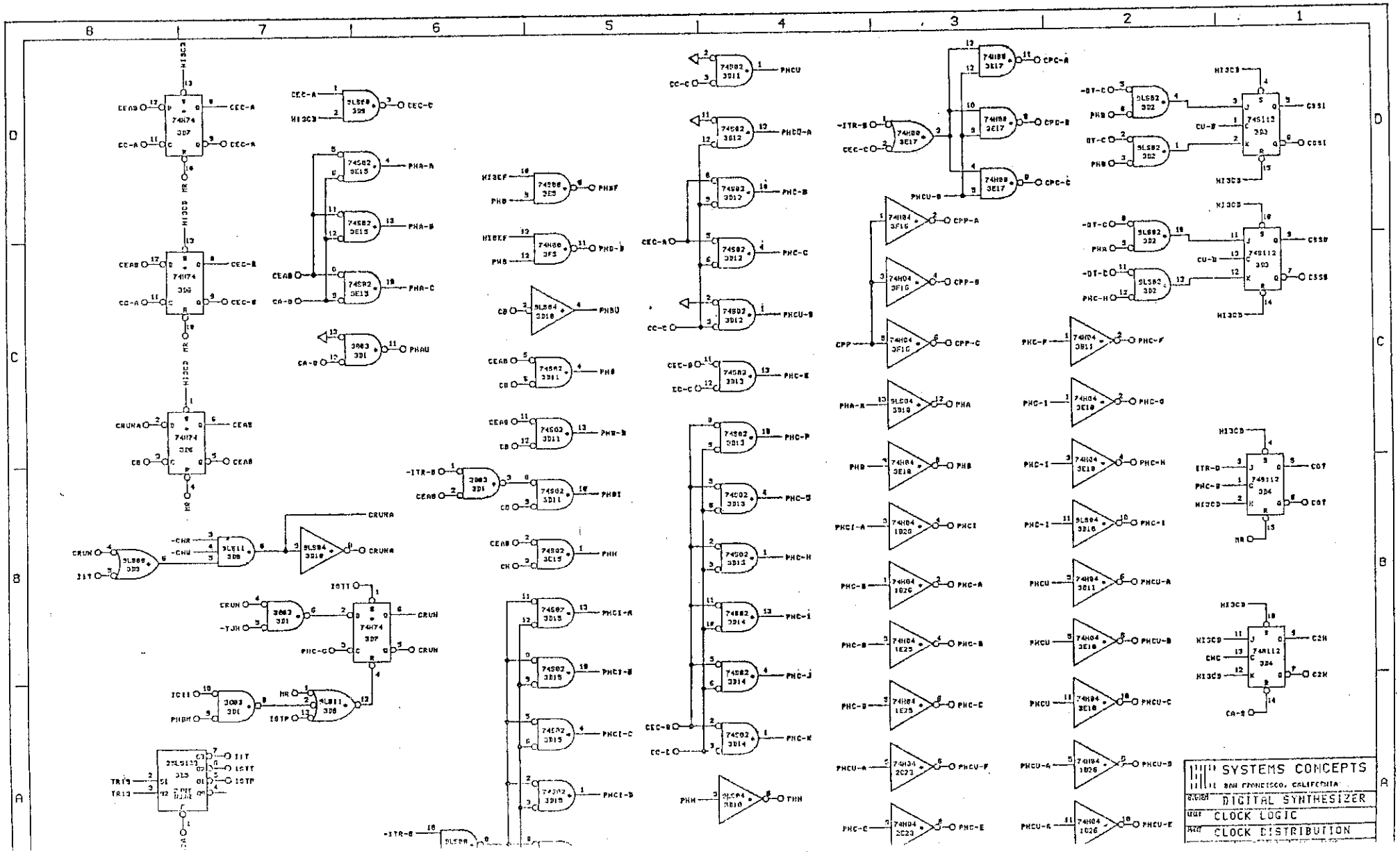
PTF=1



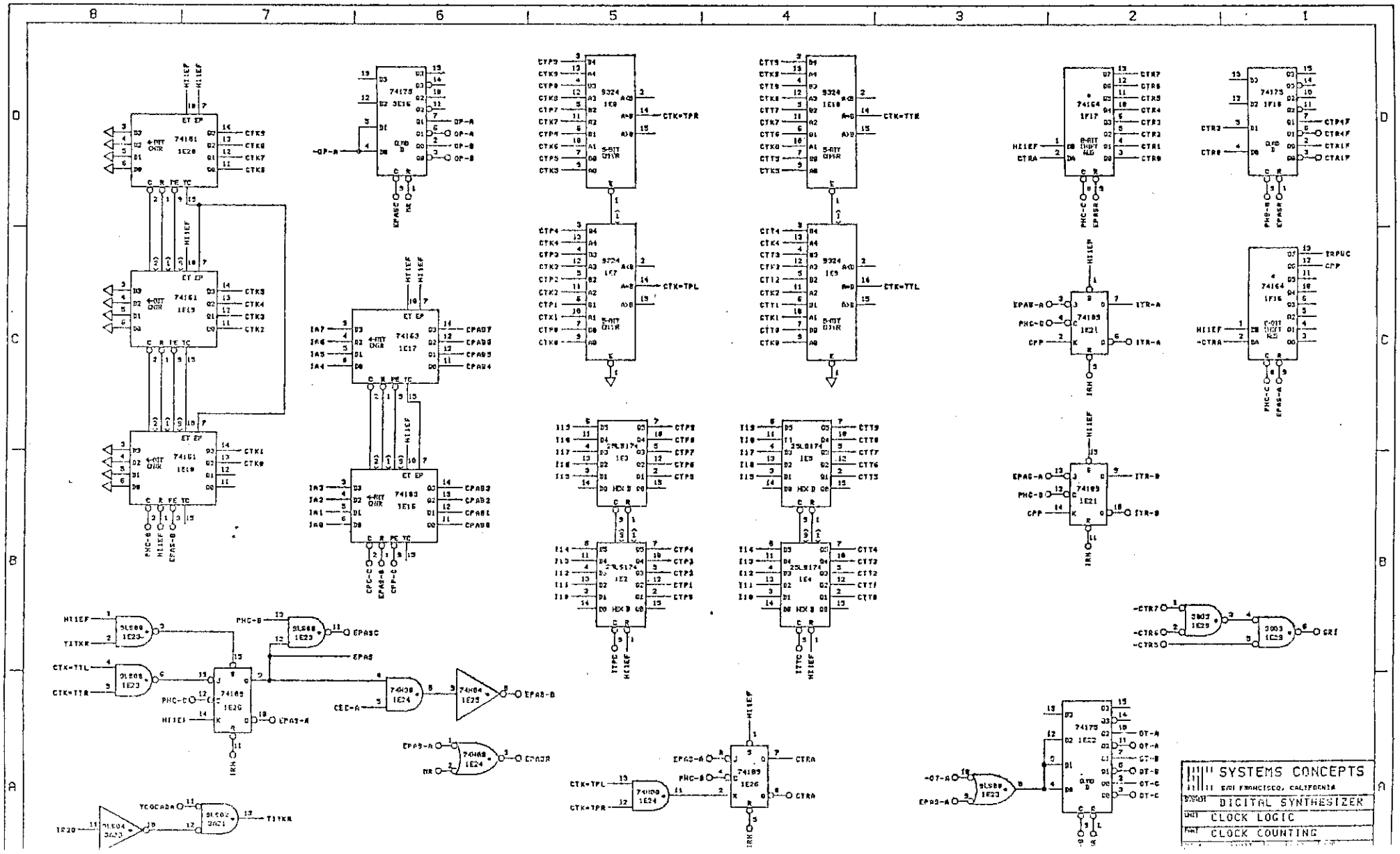
DXM; CLKGN


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
CLOCK LOGIC
CLOCK GENERATION

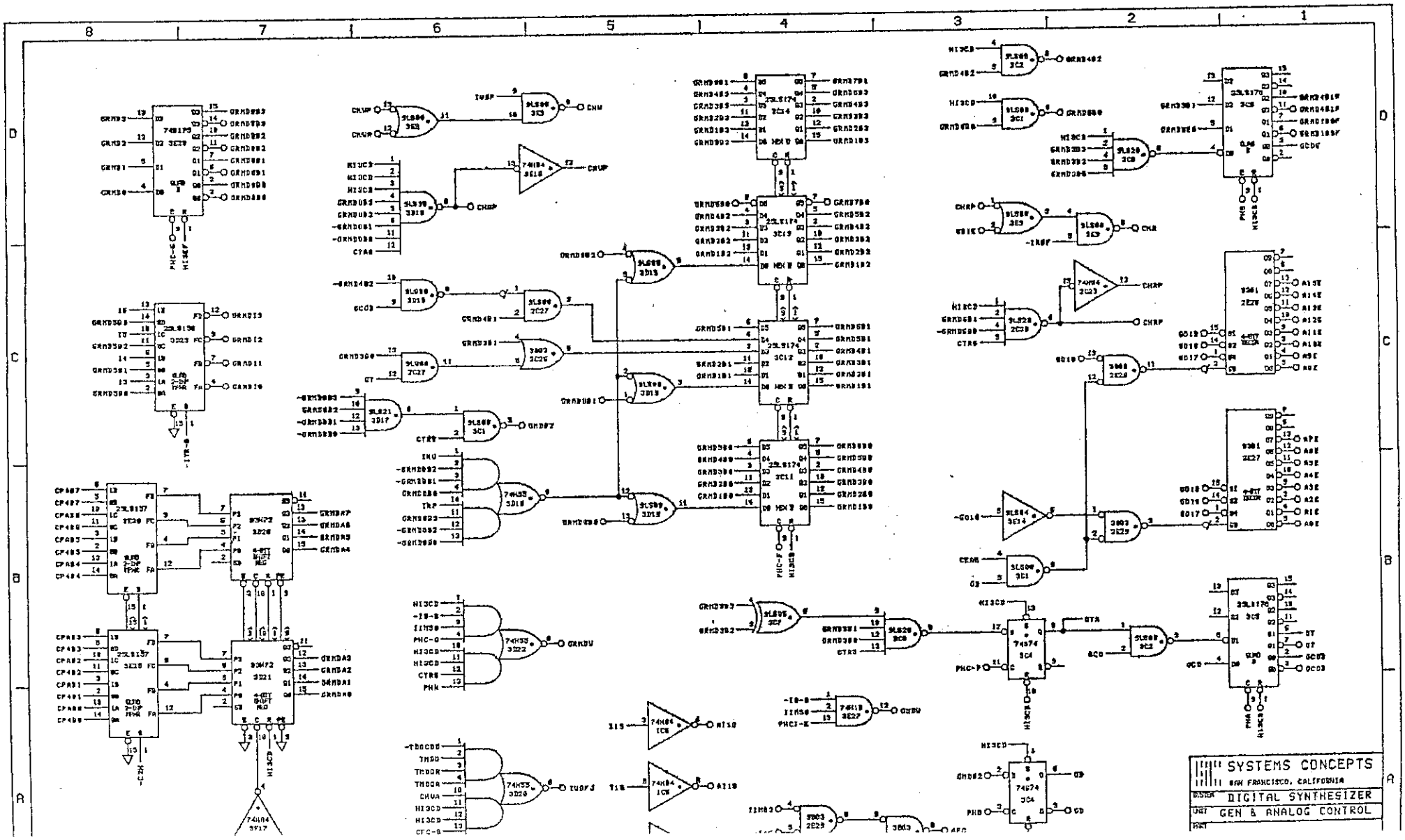
SRE; CLKDST


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
CLOCK LOGIC
CLOCK DISTRIBUTION

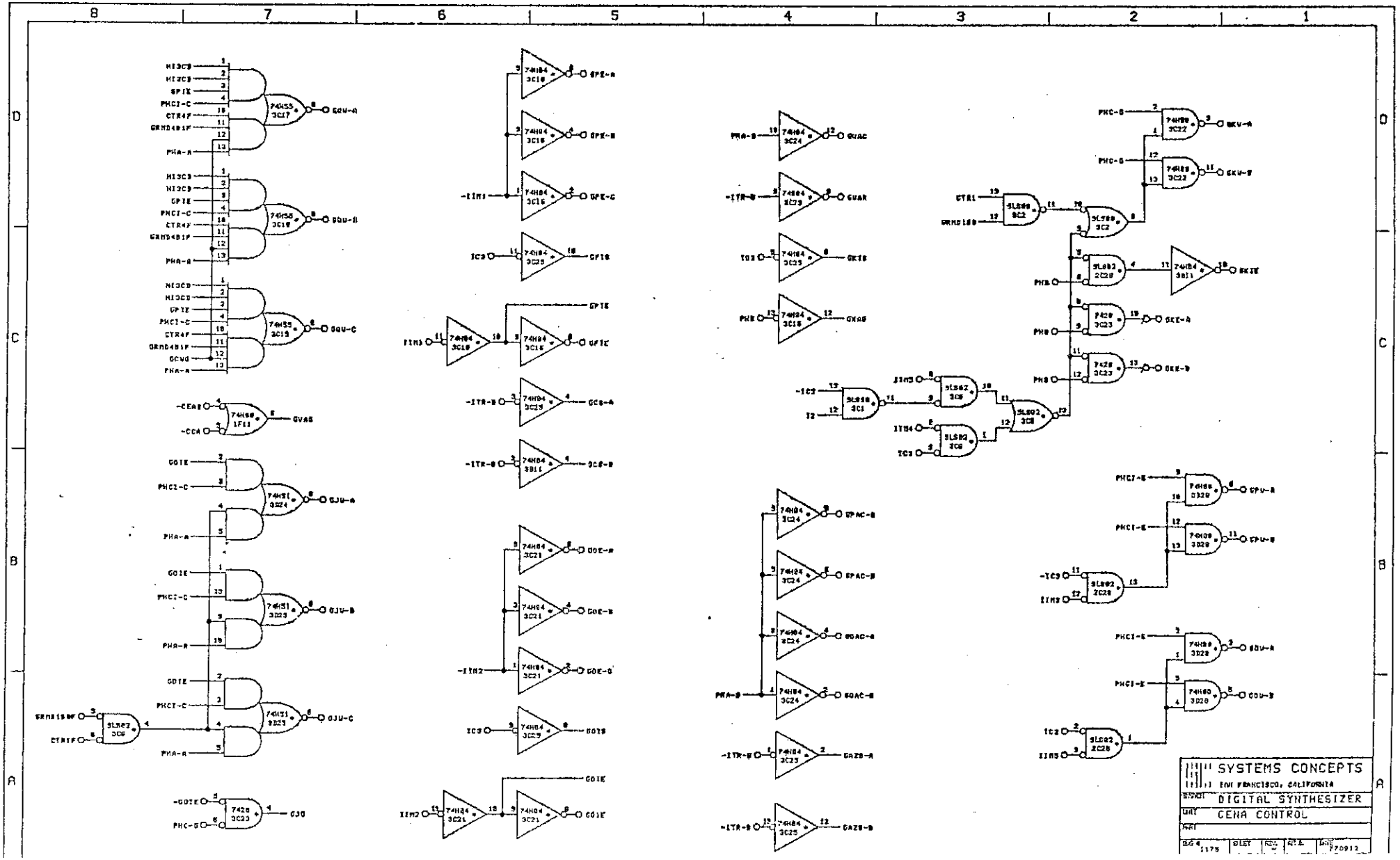
DXM; CLKCNT


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
CLOCK LOGIC
CLOCK COUNTING

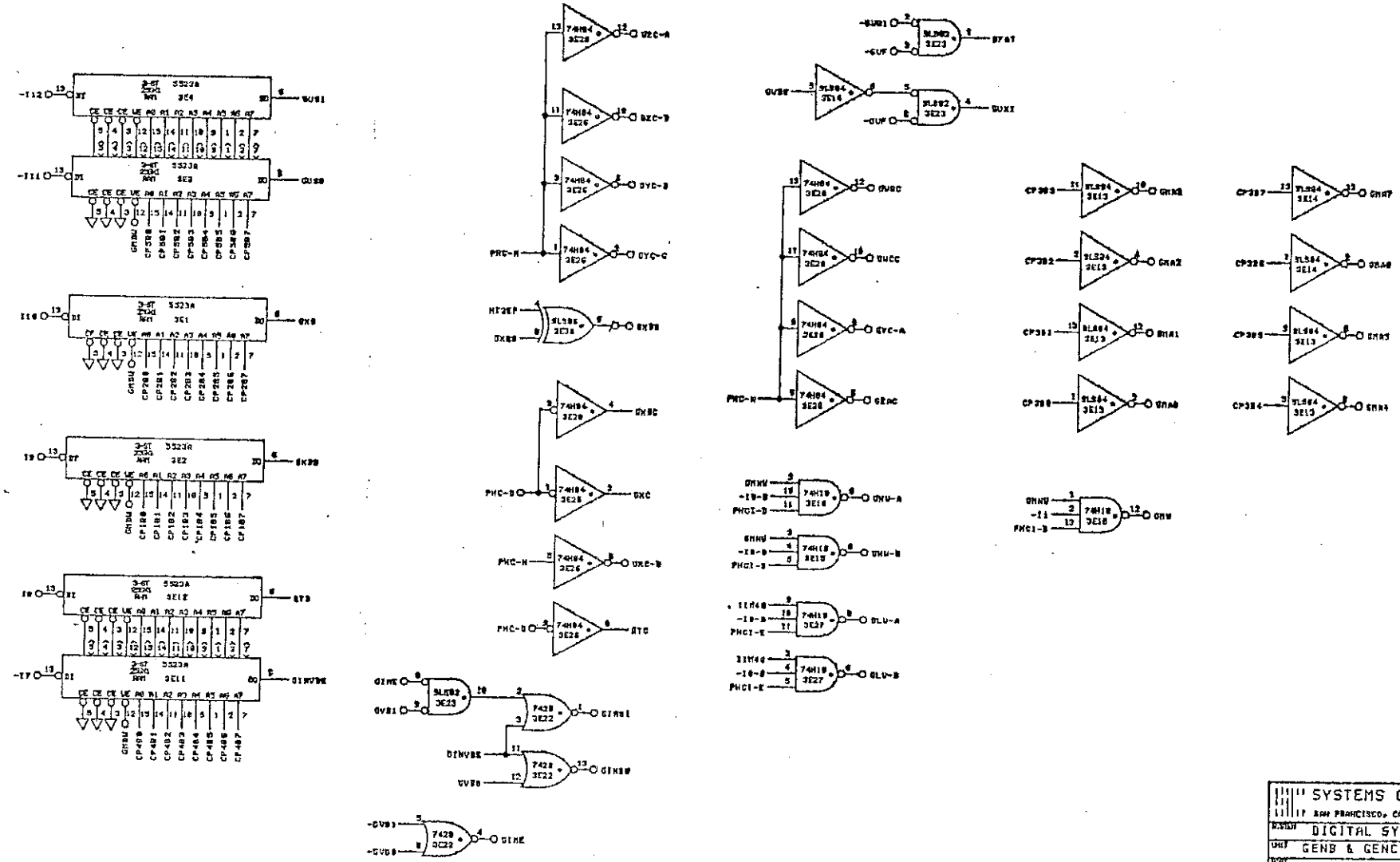
DXM, GENANC


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
GEN & ANALOG CONTROL

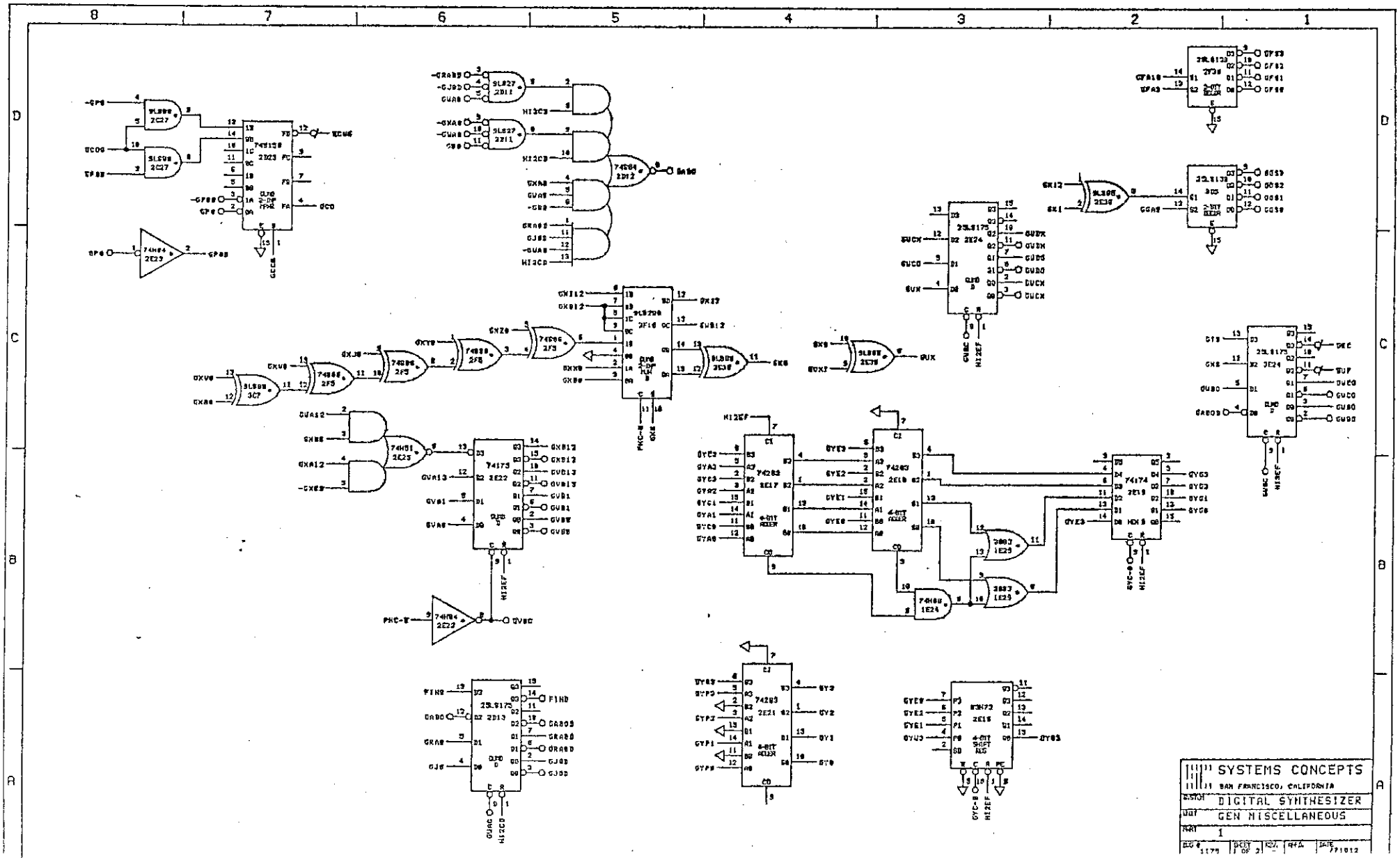
BG; GENAC


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
CENA CONTROL

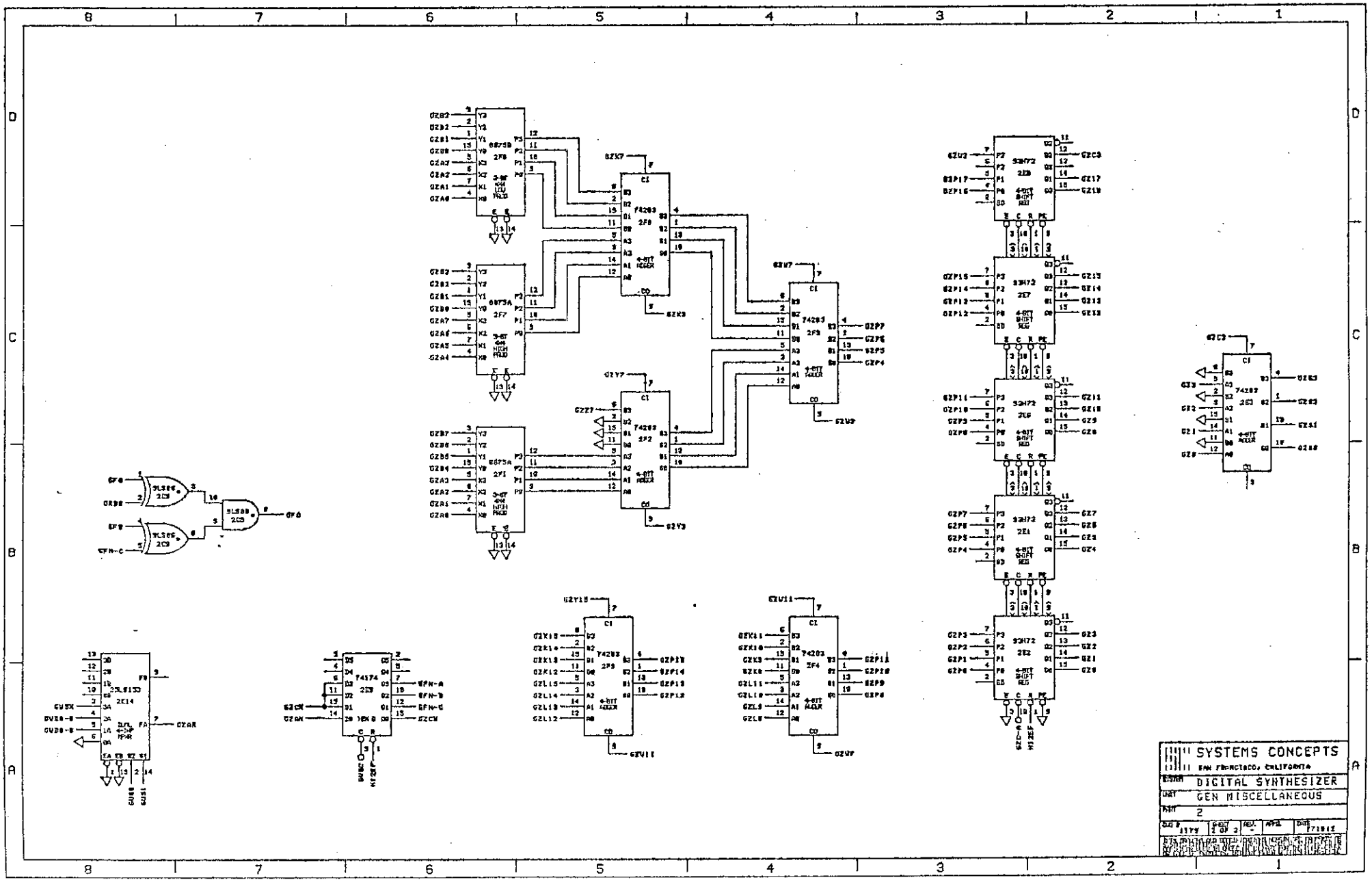
SRE; GENBCC


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
GENB & GENC CONTROL

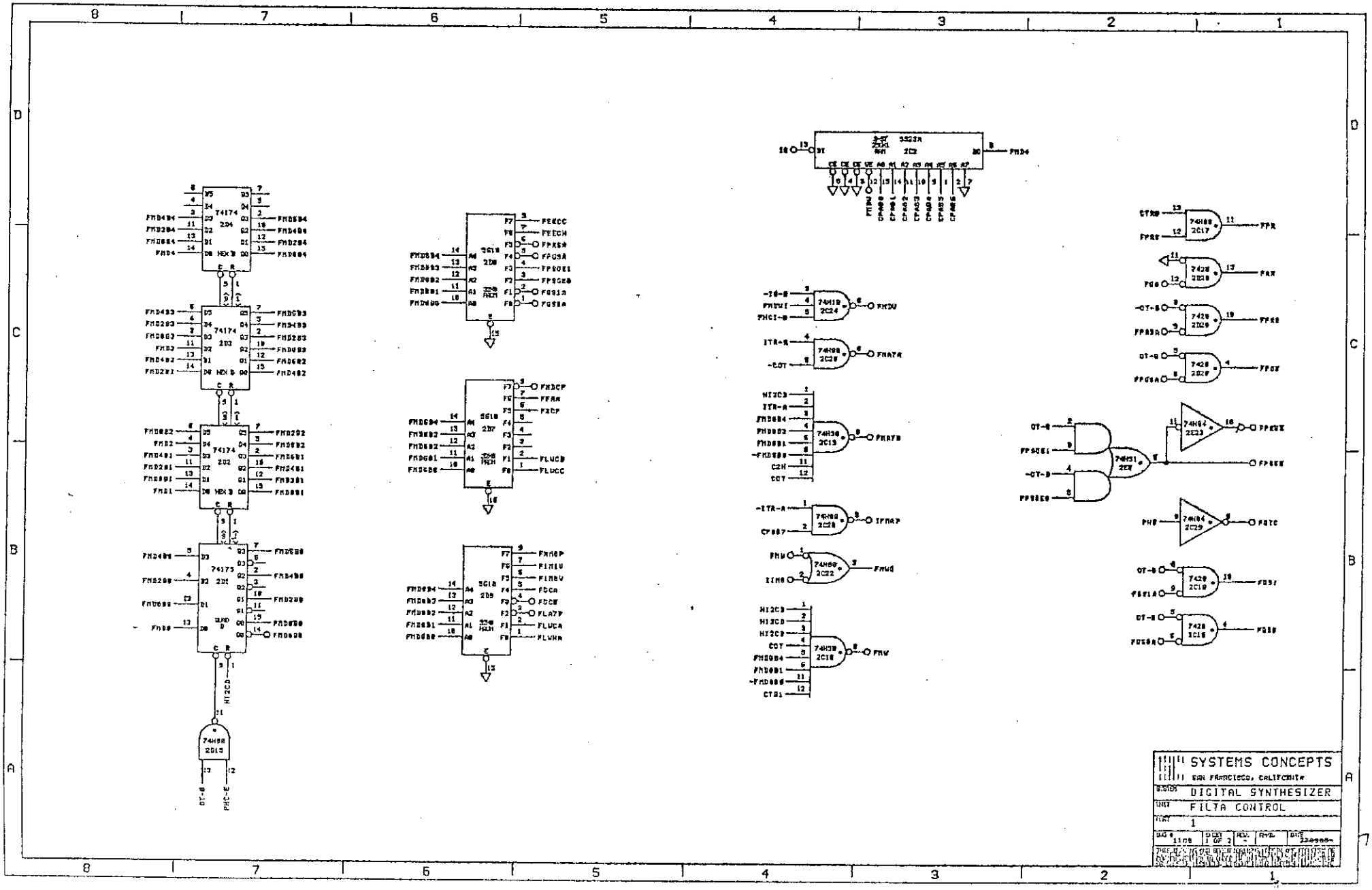
SWB; GENM1


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
GEN MISCELLANEOUS
1

SWB; GENM2

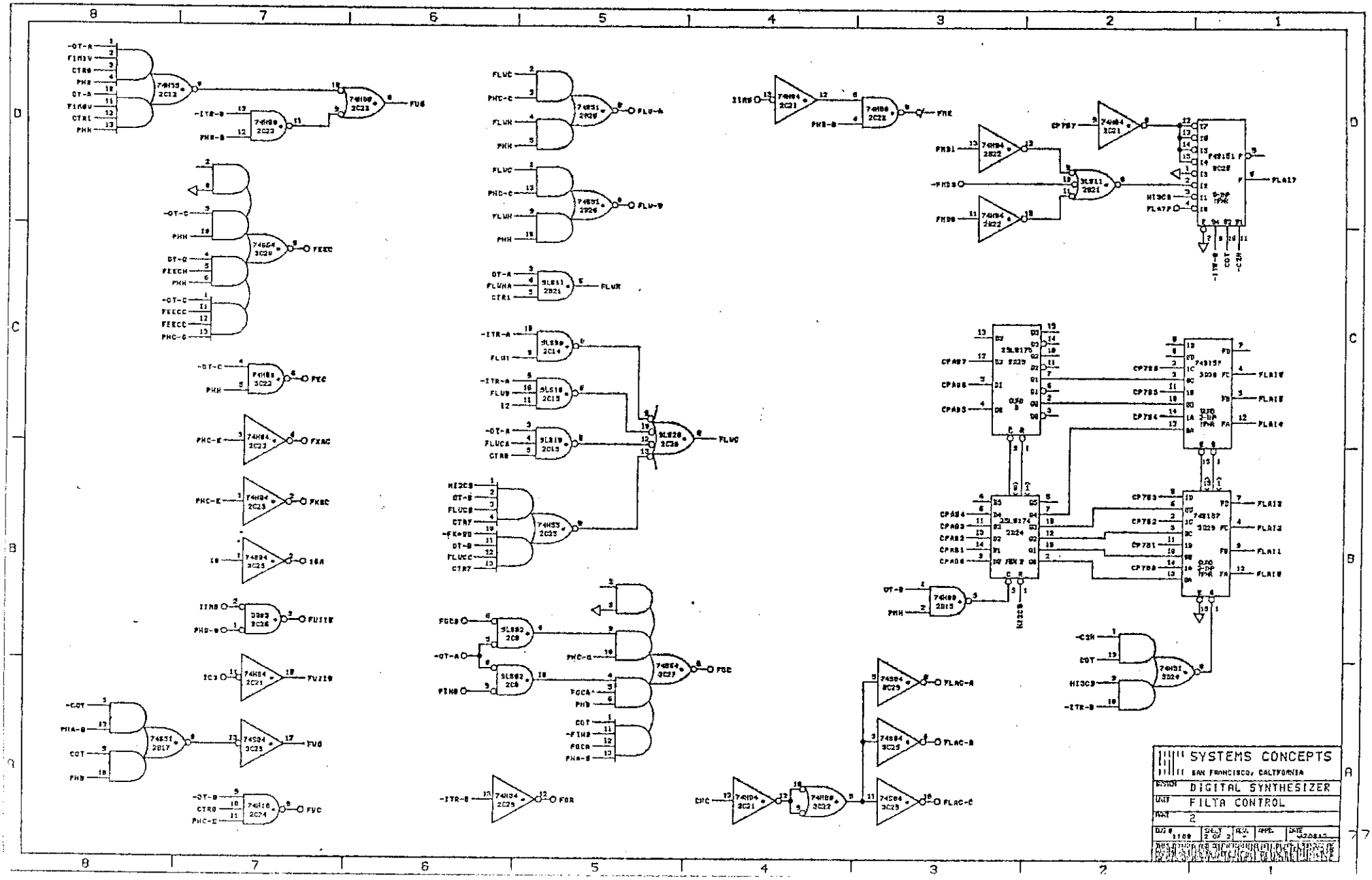

SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
GEN MISCELLANEOUS
2
1 OF 2

DXM; FILTAC1

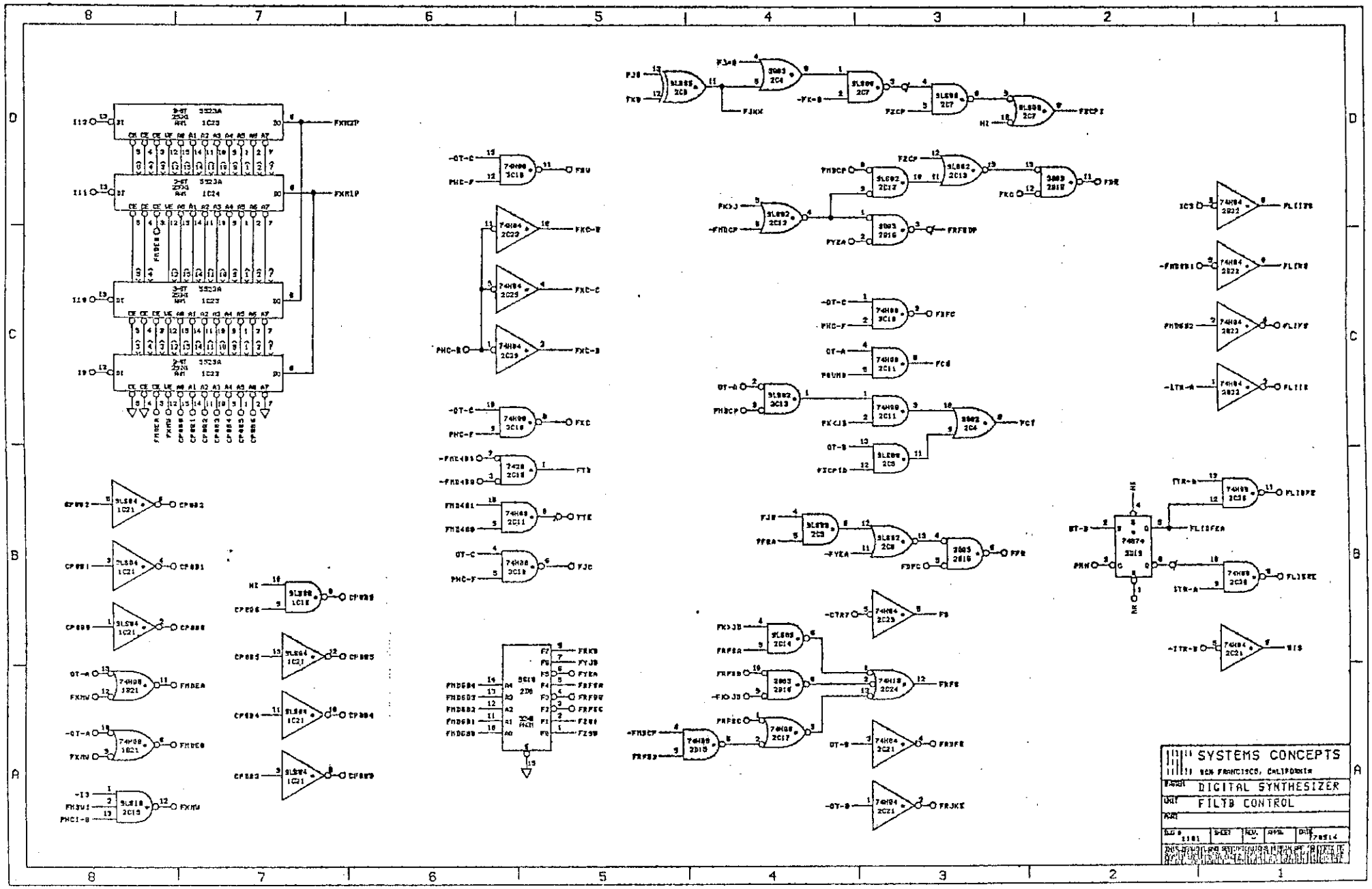



771

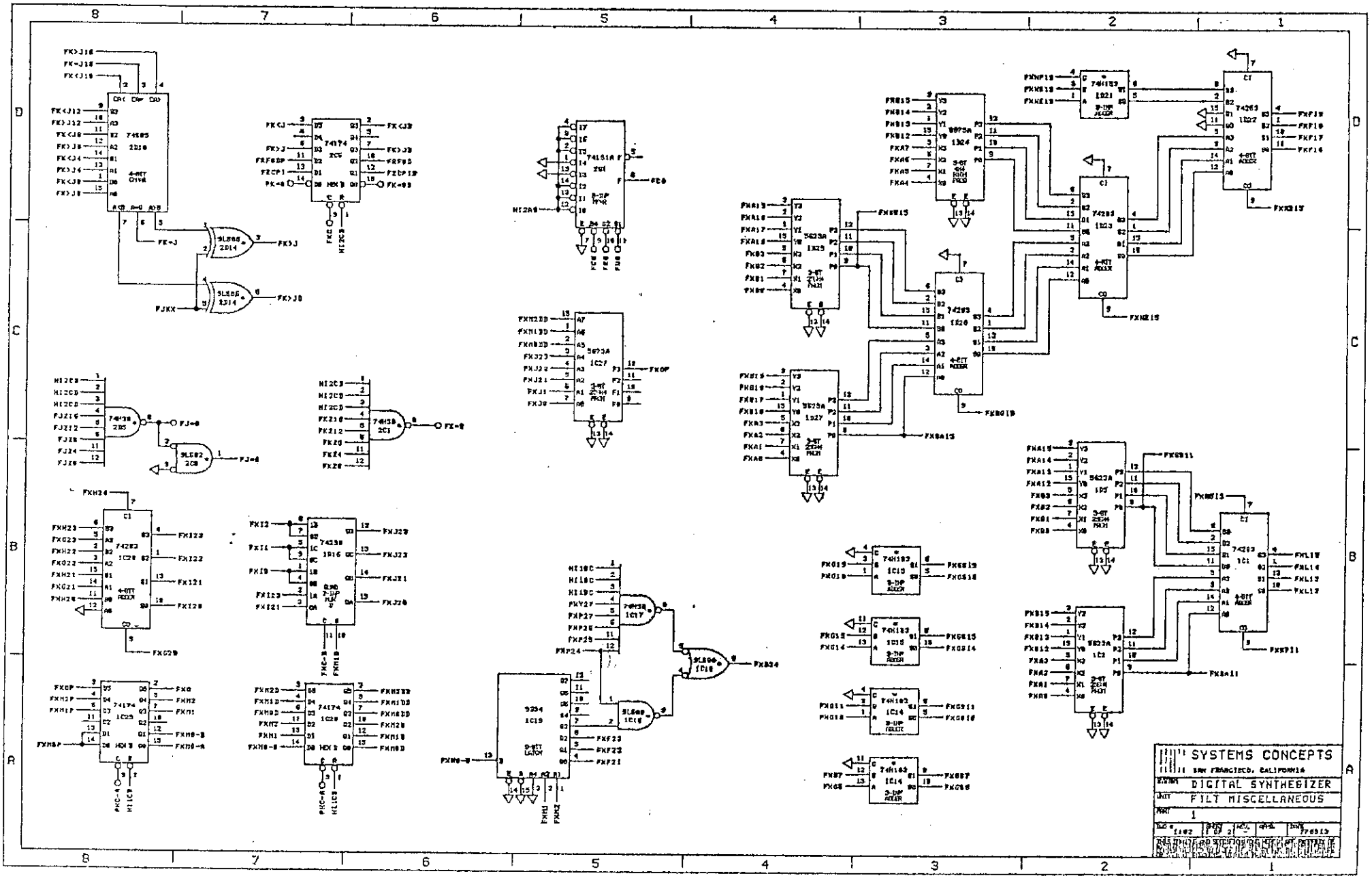
DXM; FLTAC2


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
FILTA CONTROL
2

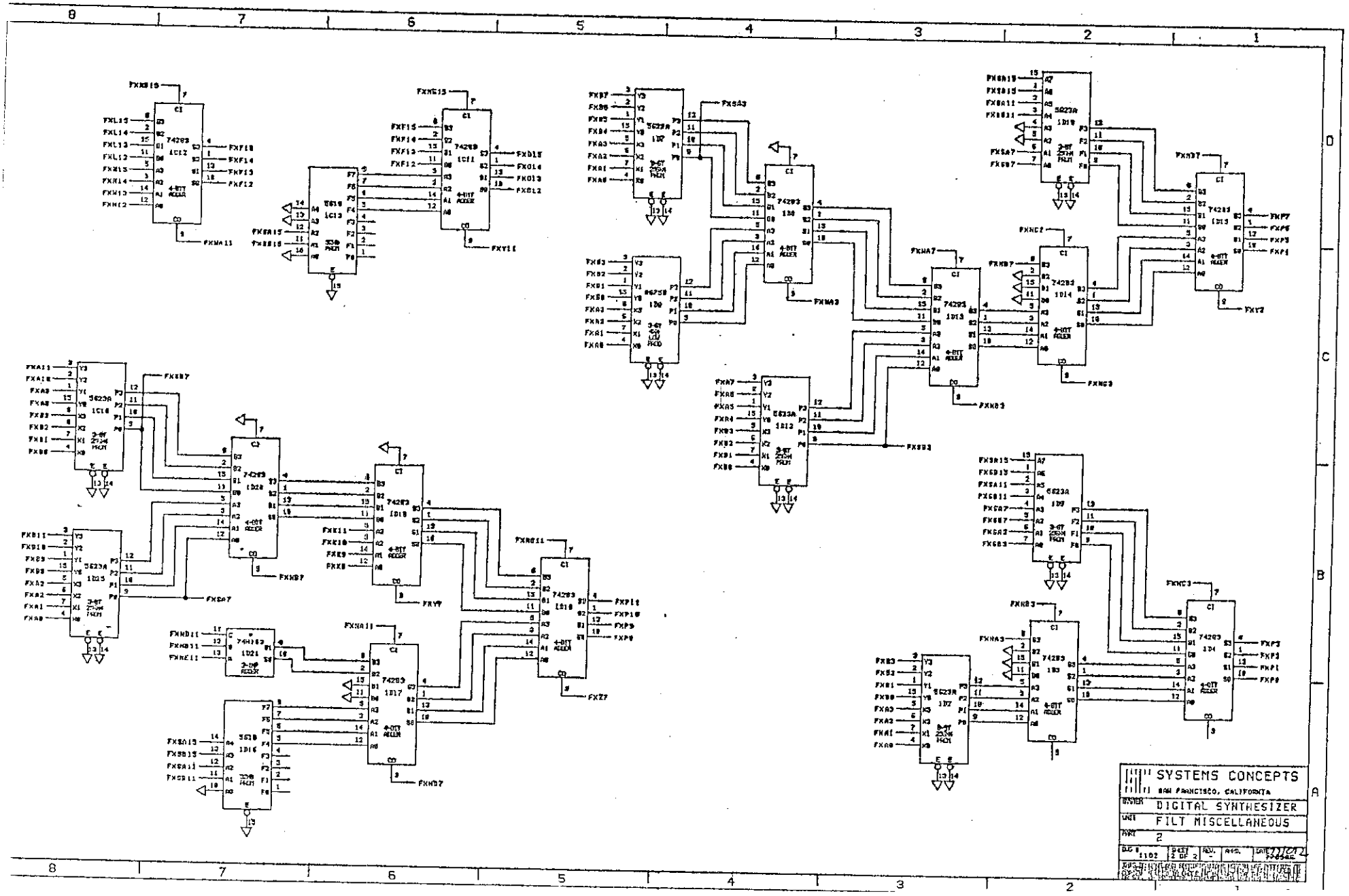
DXM; FILT BC


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
FILTB CONTROL

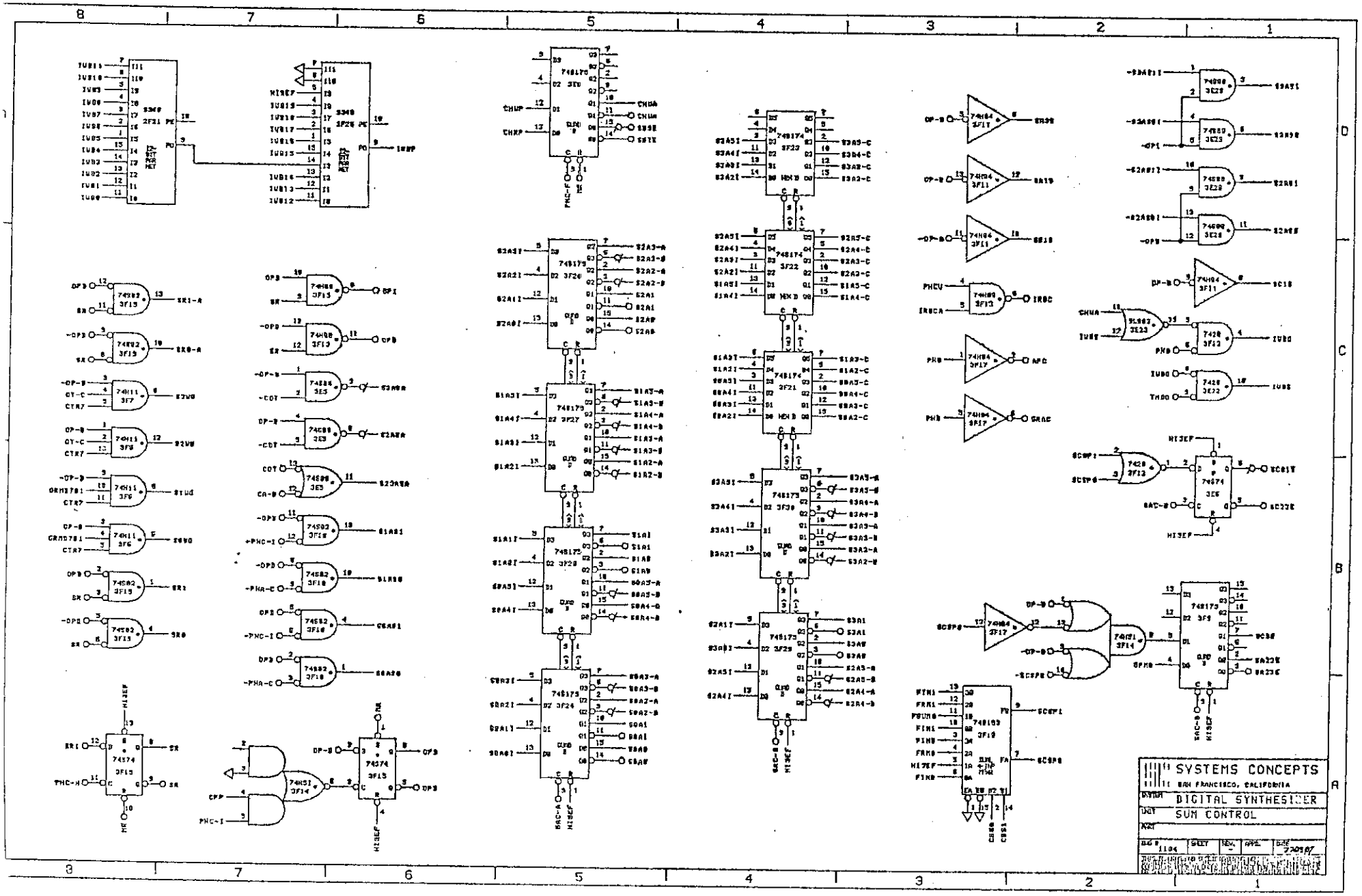
DXM;FILTM1


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
FILT MISCELLANEOUS
1

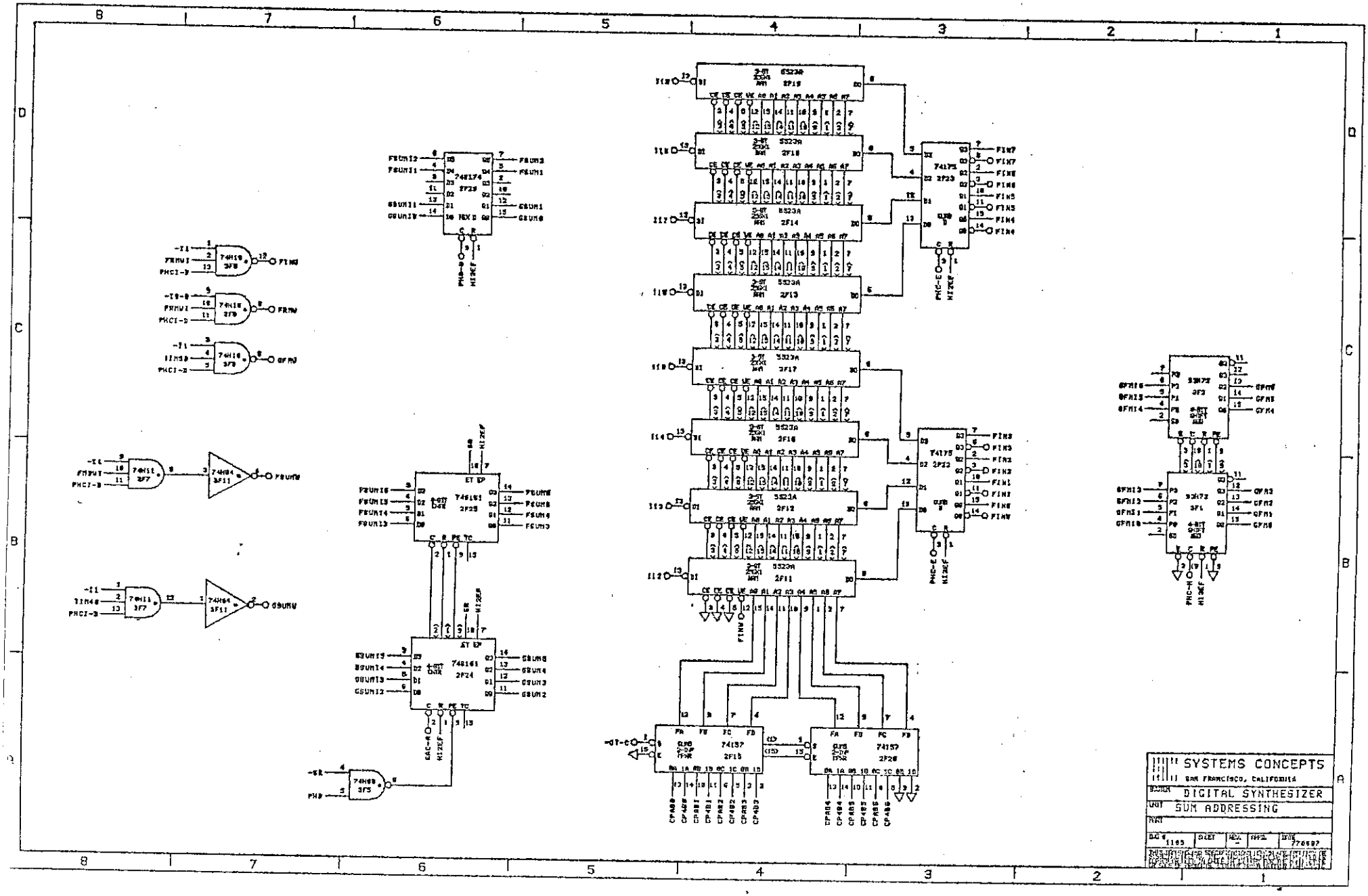
DXM; FILT/M2


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
FILT MISCELLANEOUS
2

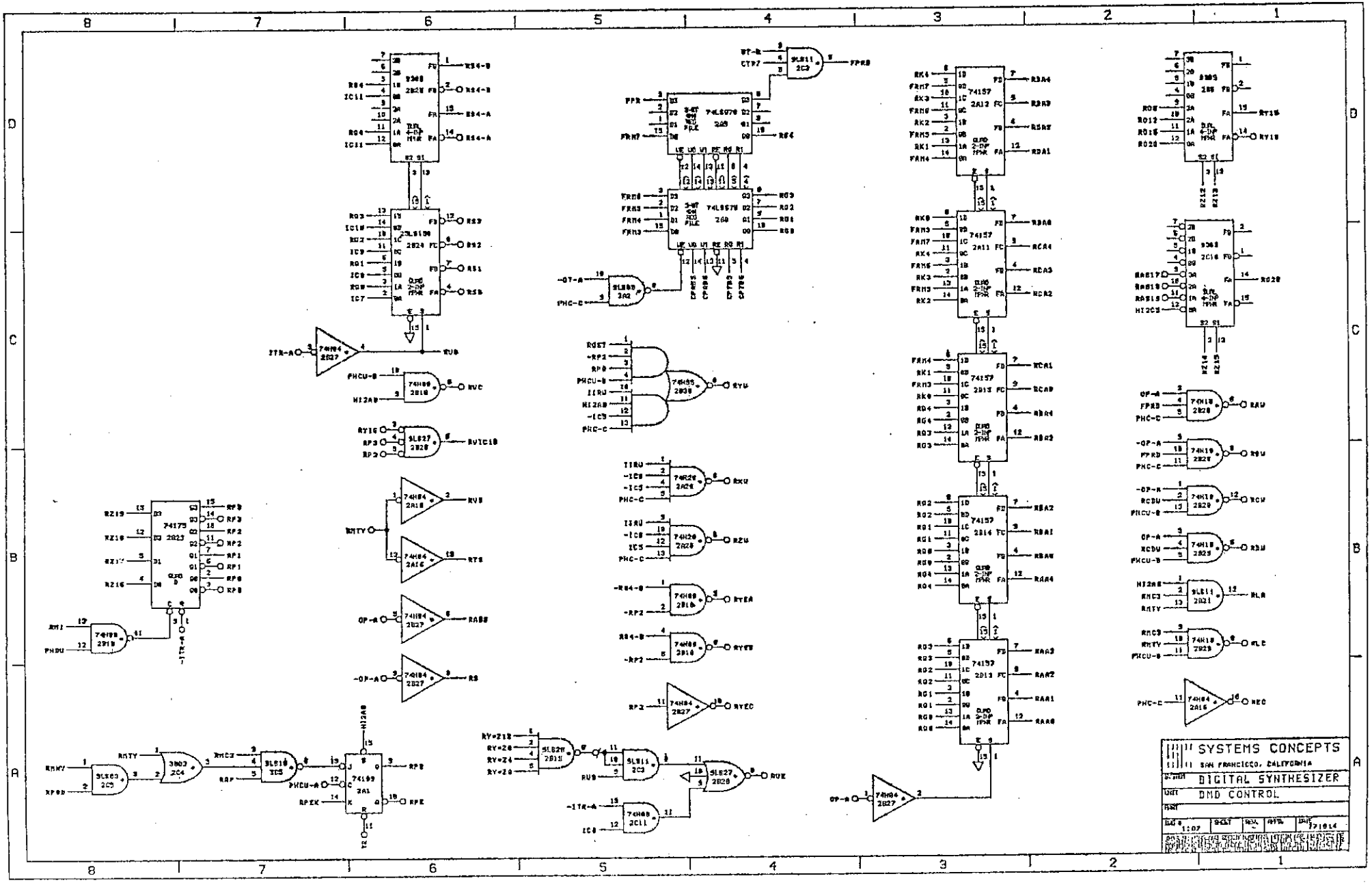
SWB; SUM C


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
SUM CONTROL

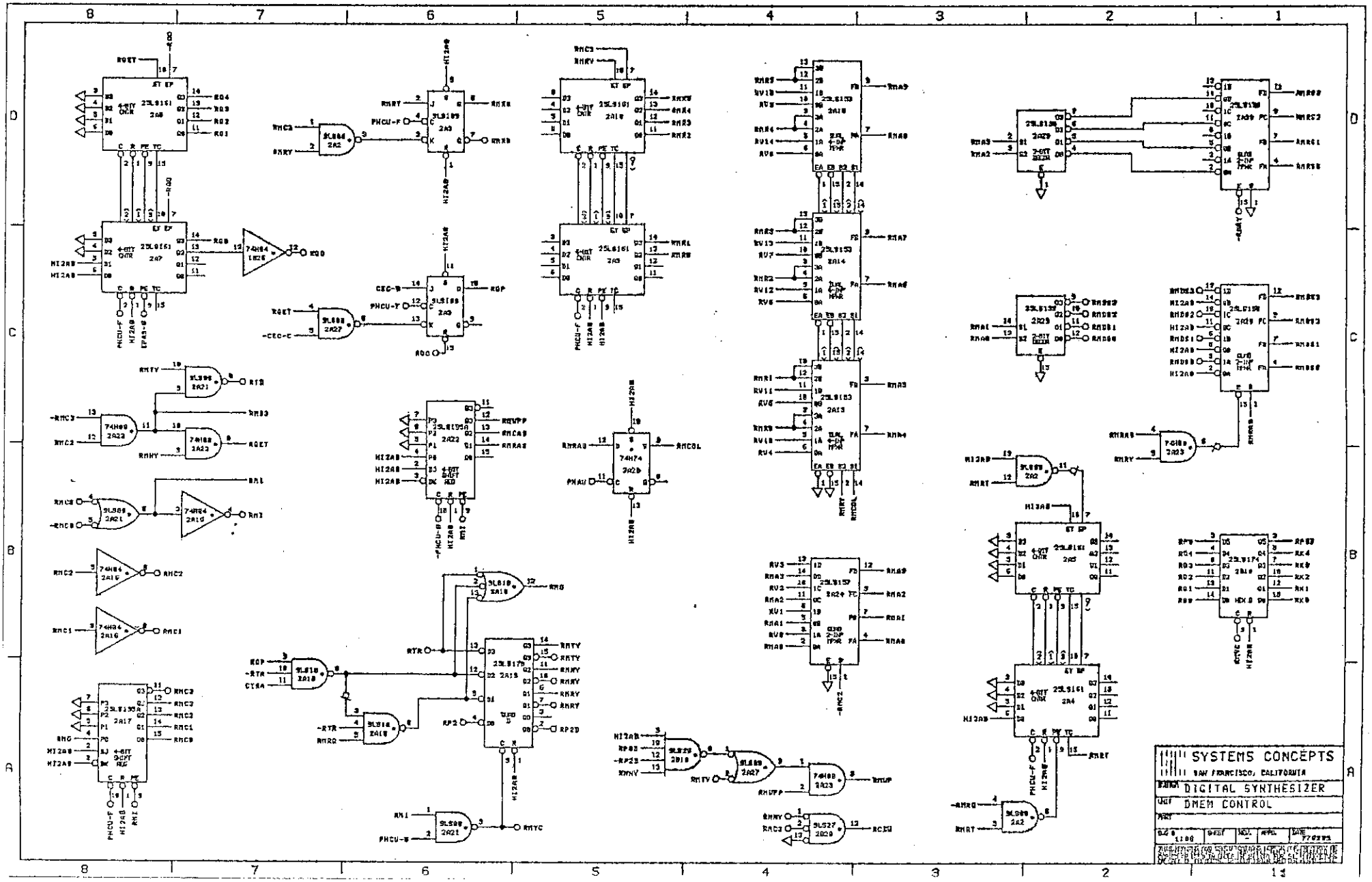
DXM; SUMADD


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
SUM ADDRESSING

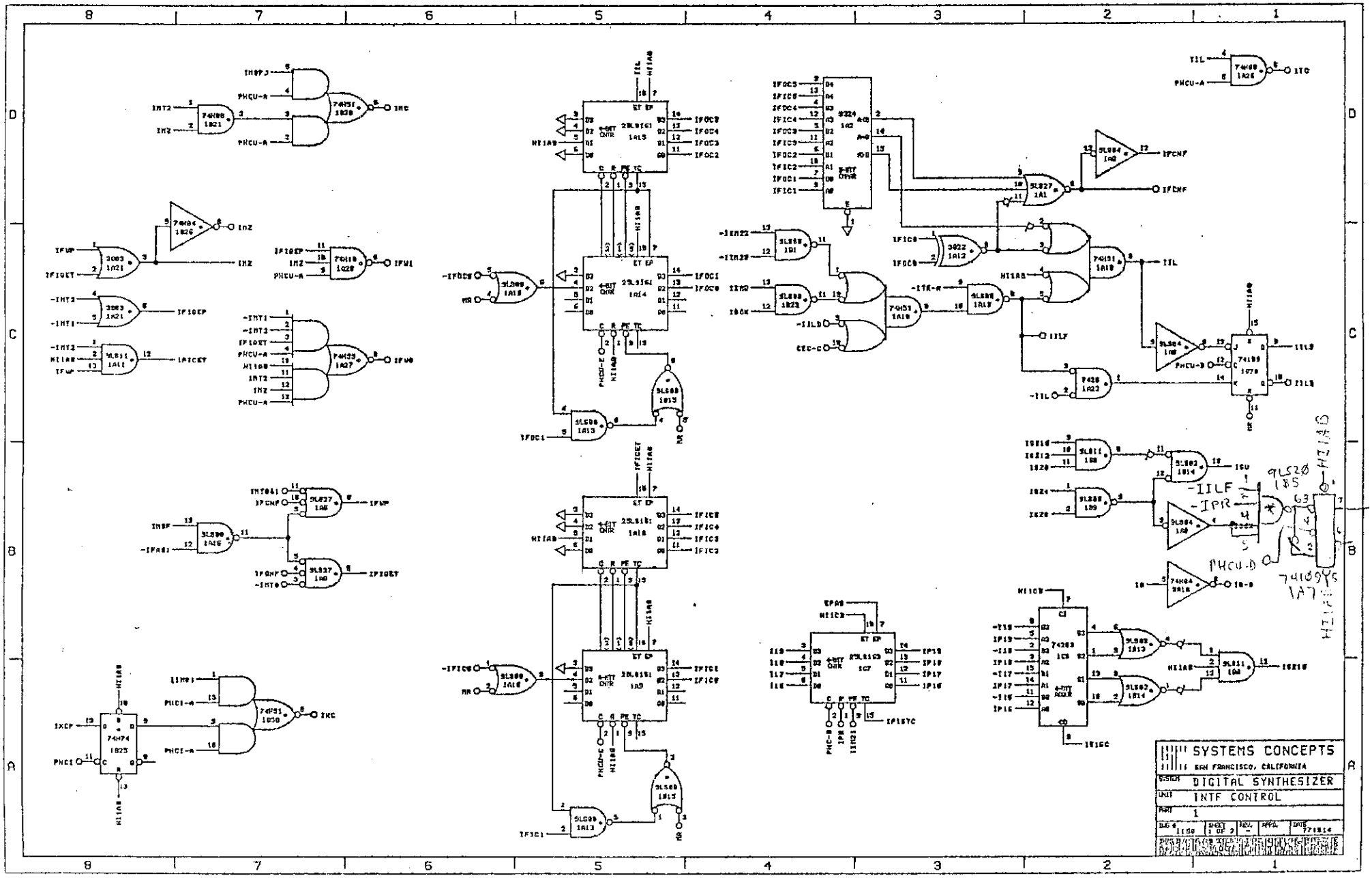
BG; DMDC


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
DMD CONTROL

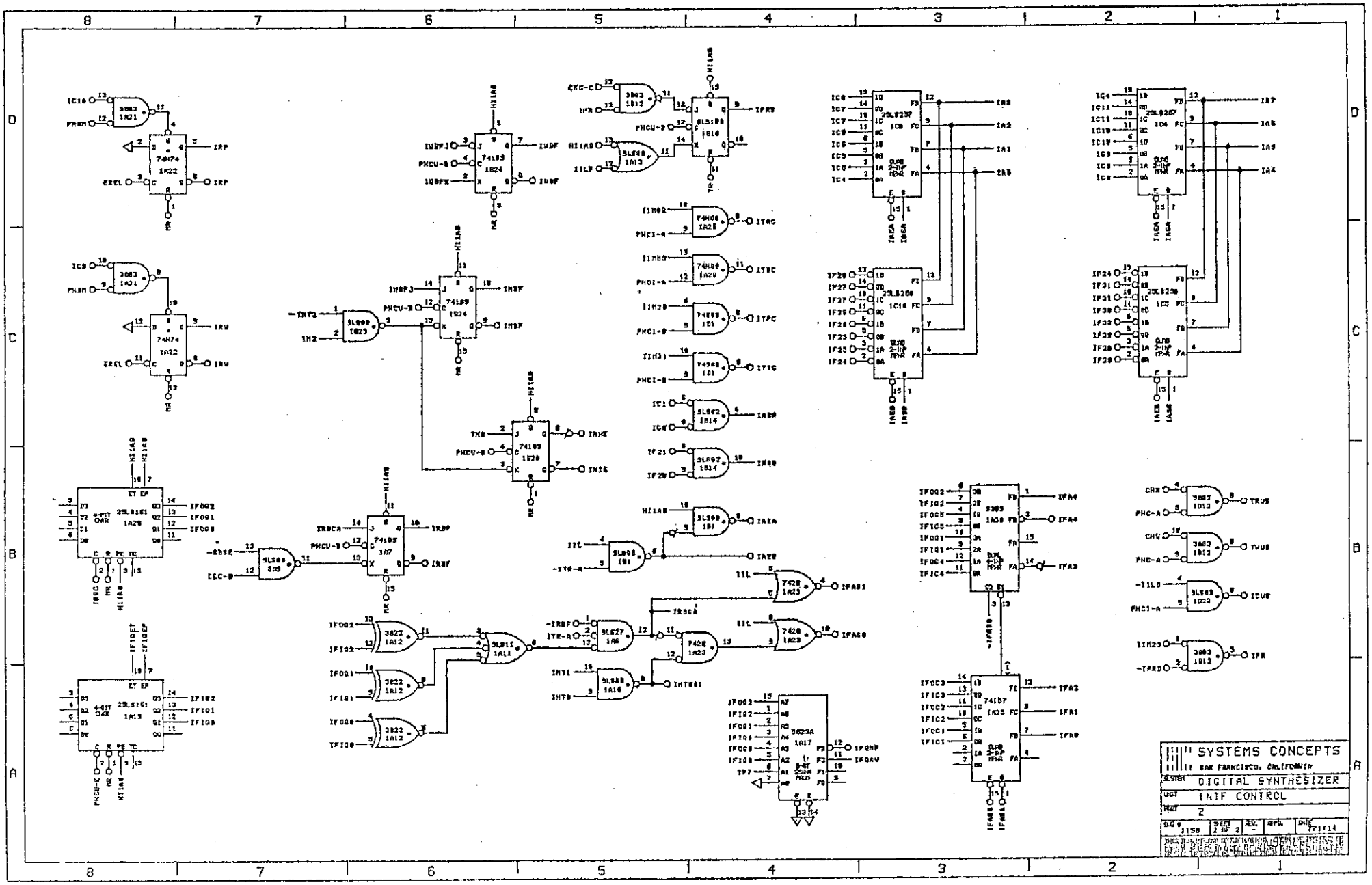
SRE; DMEMC


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
DMEM CONTROL

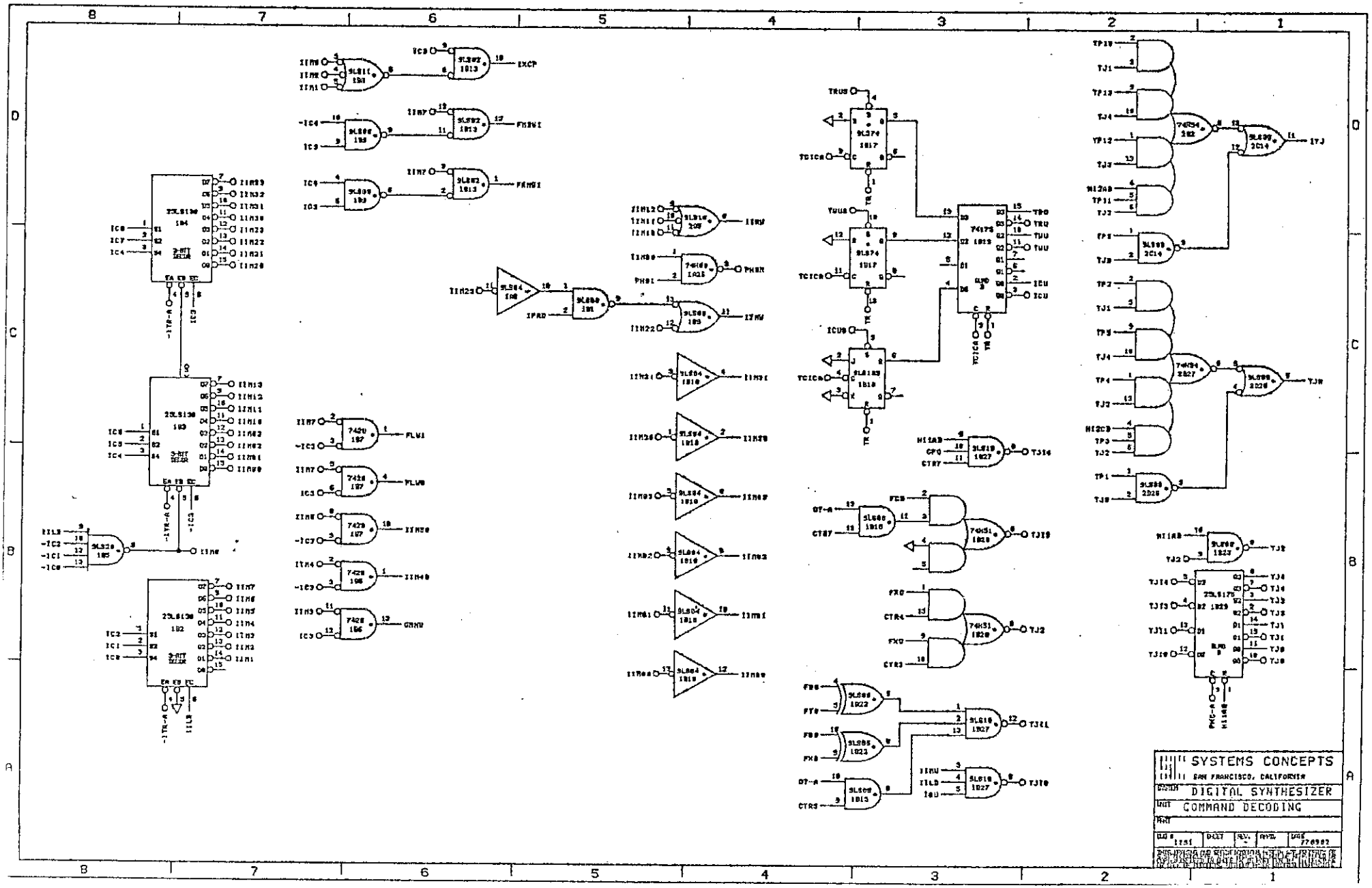
SWB; INTFC1


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
INTF CONTROL
1
1 OF 2
9LS20
1B5
74LS95
1A7
ISO
-ILLF
-IPR
PHCU-D

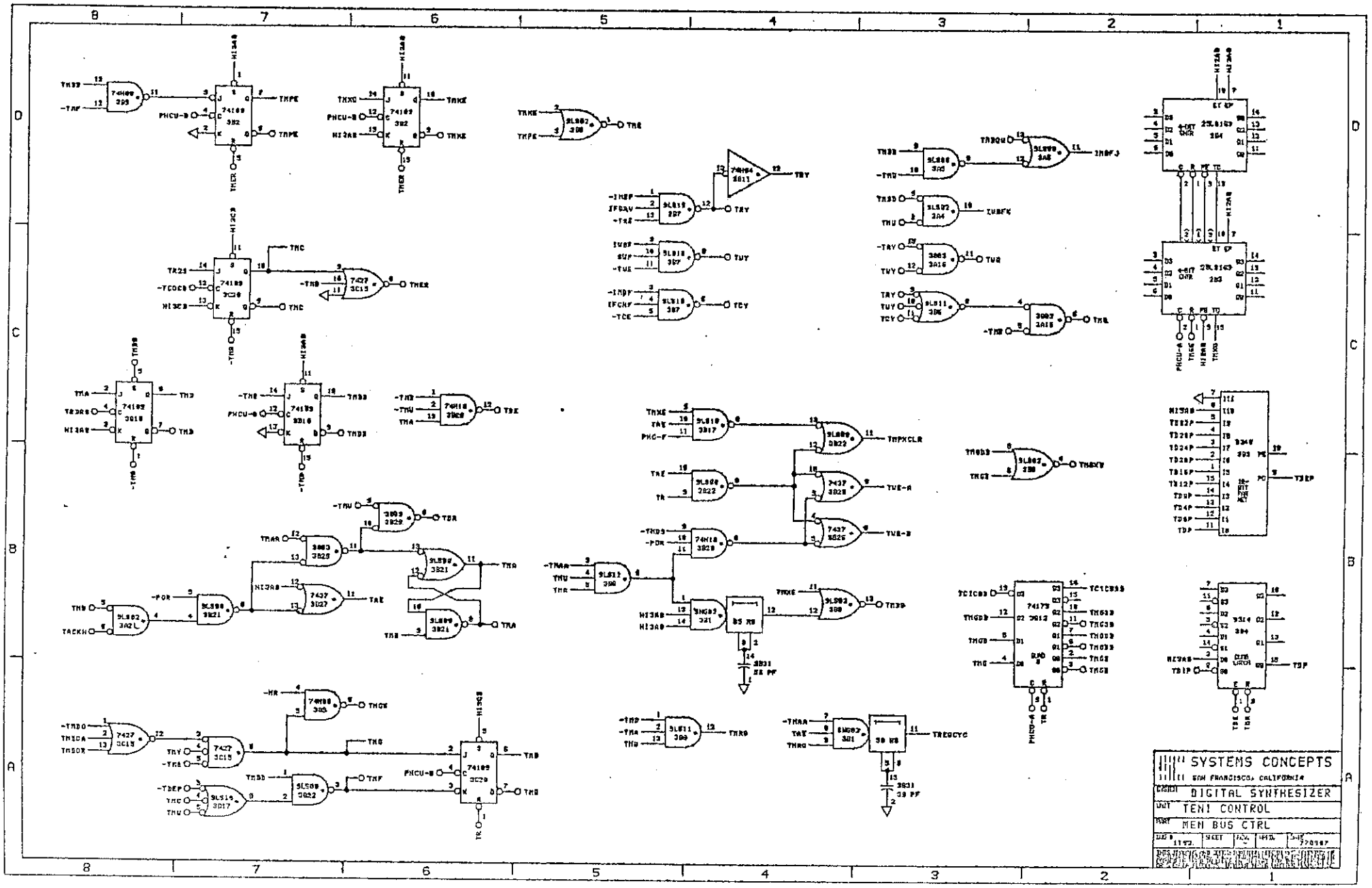
SWB; INTFC2


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
INTF CONTROL
2

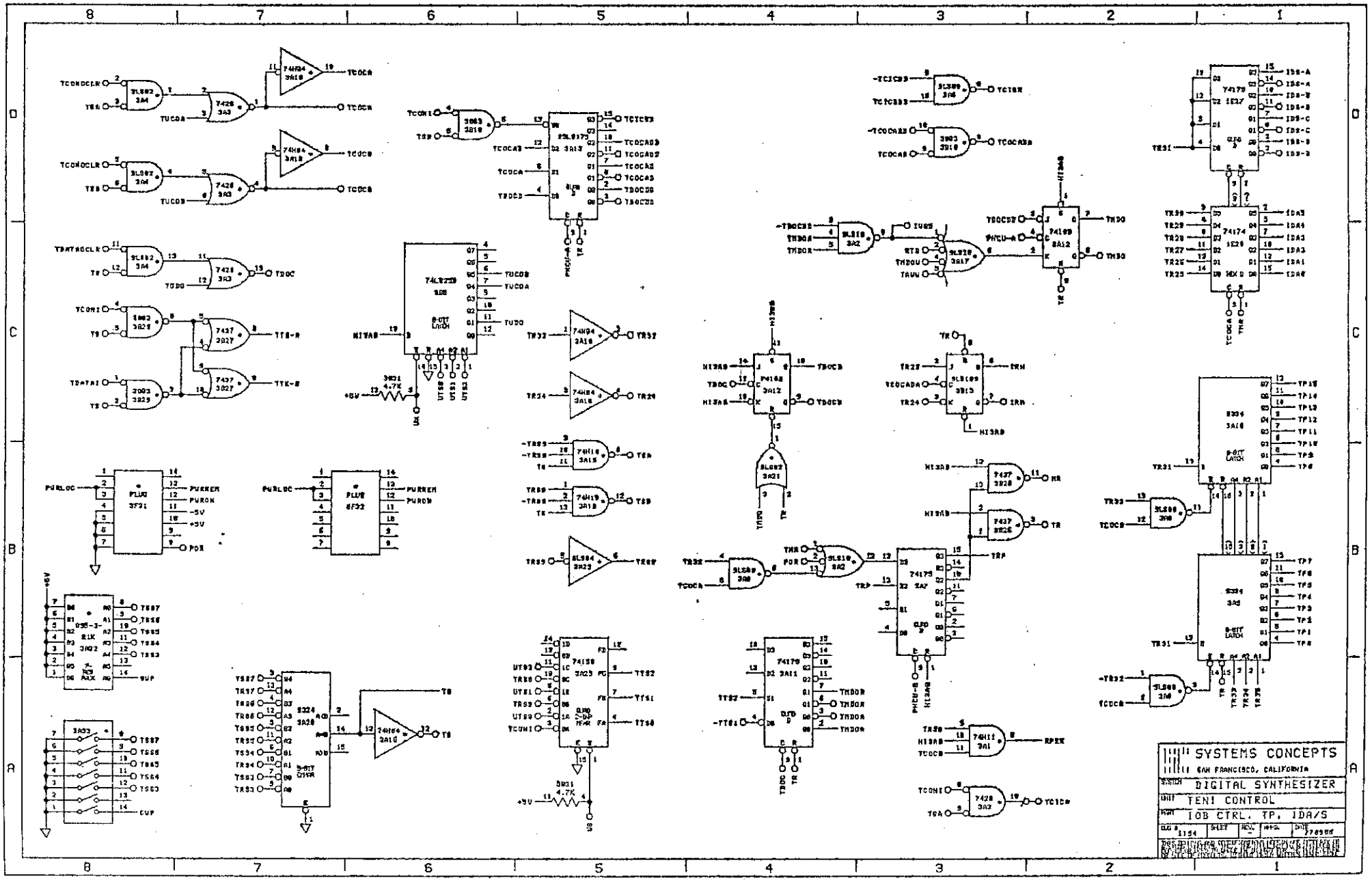
SWB; COMDEC


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
COMMAND DECODING

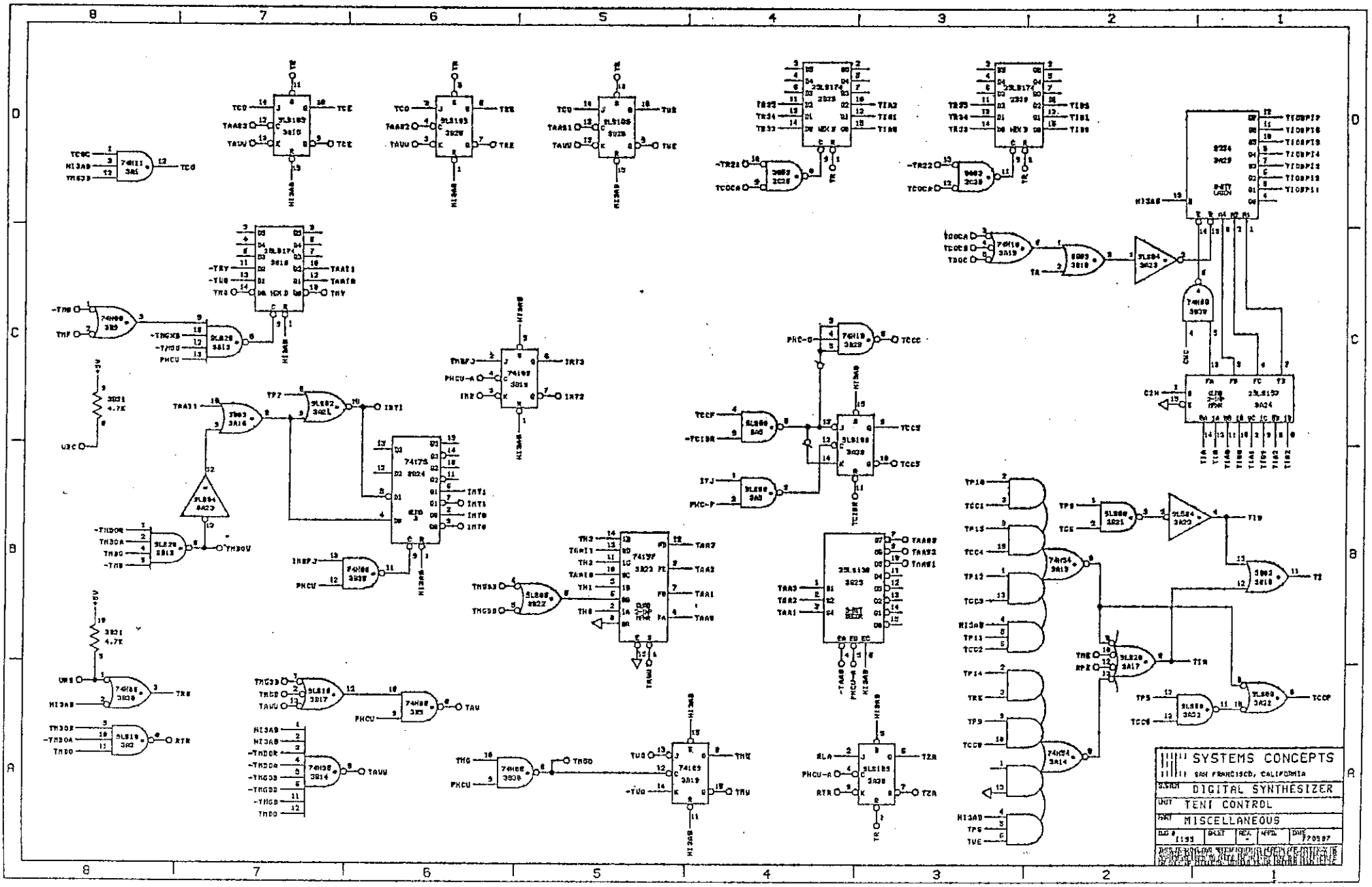
BG; MEMBC


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
TENI CONTROL
MEM BUS CTRL

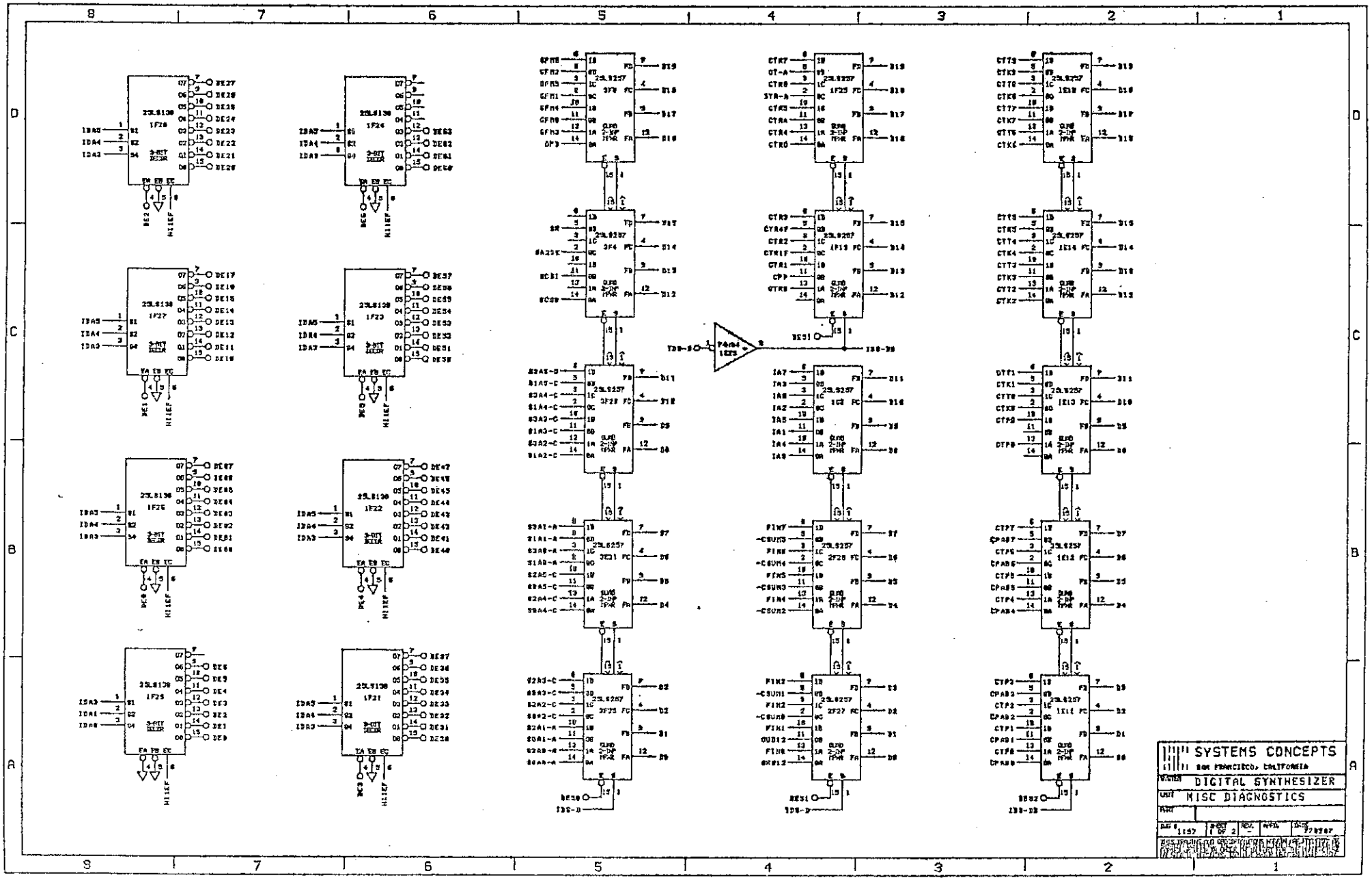
SWB; IOBC


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
TENI CONTROL
IOB CTRL, TP, IDA/S

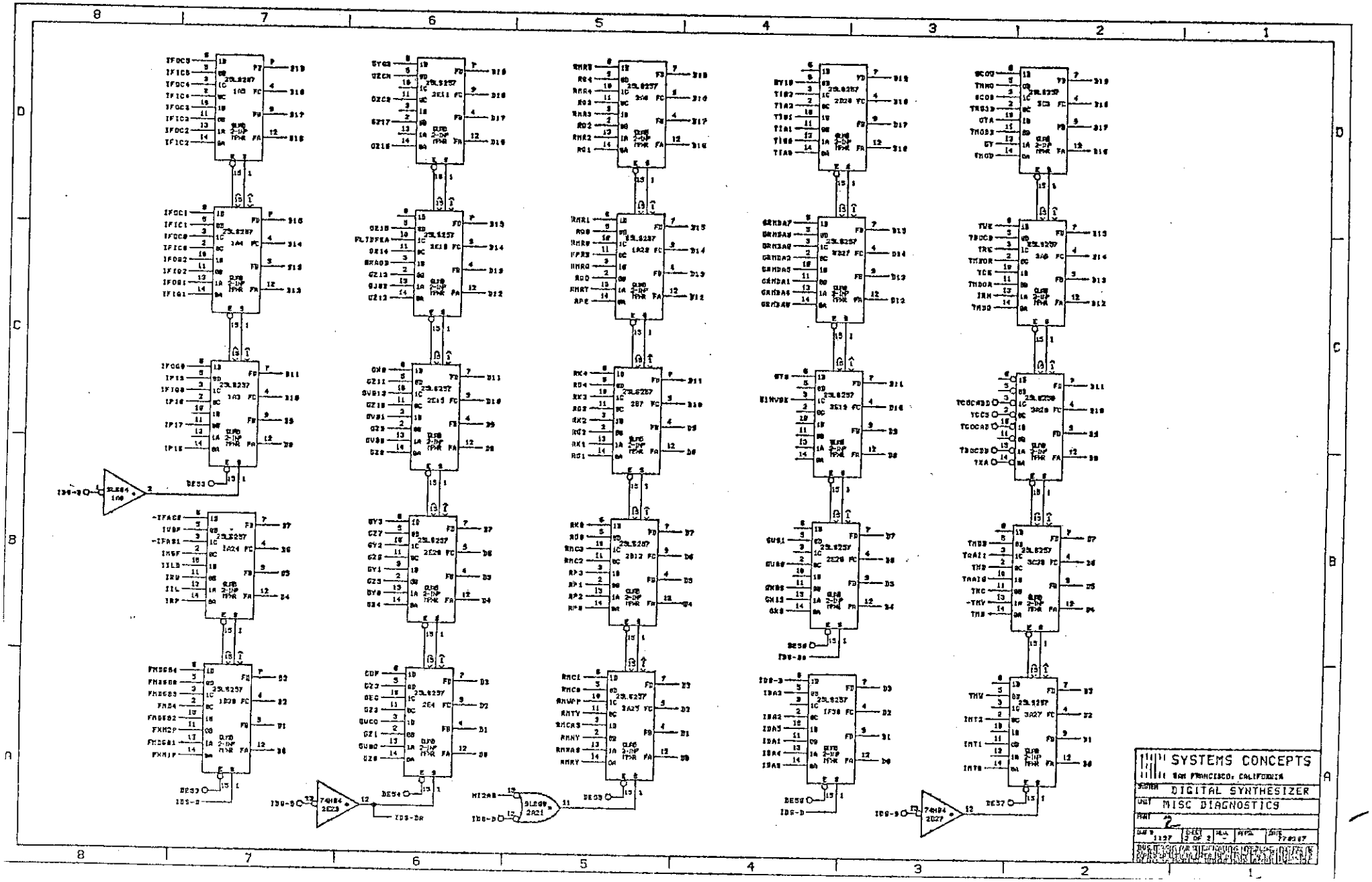
SRE;TMISC


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
TENI CONTROL
MISCELLANEOUS

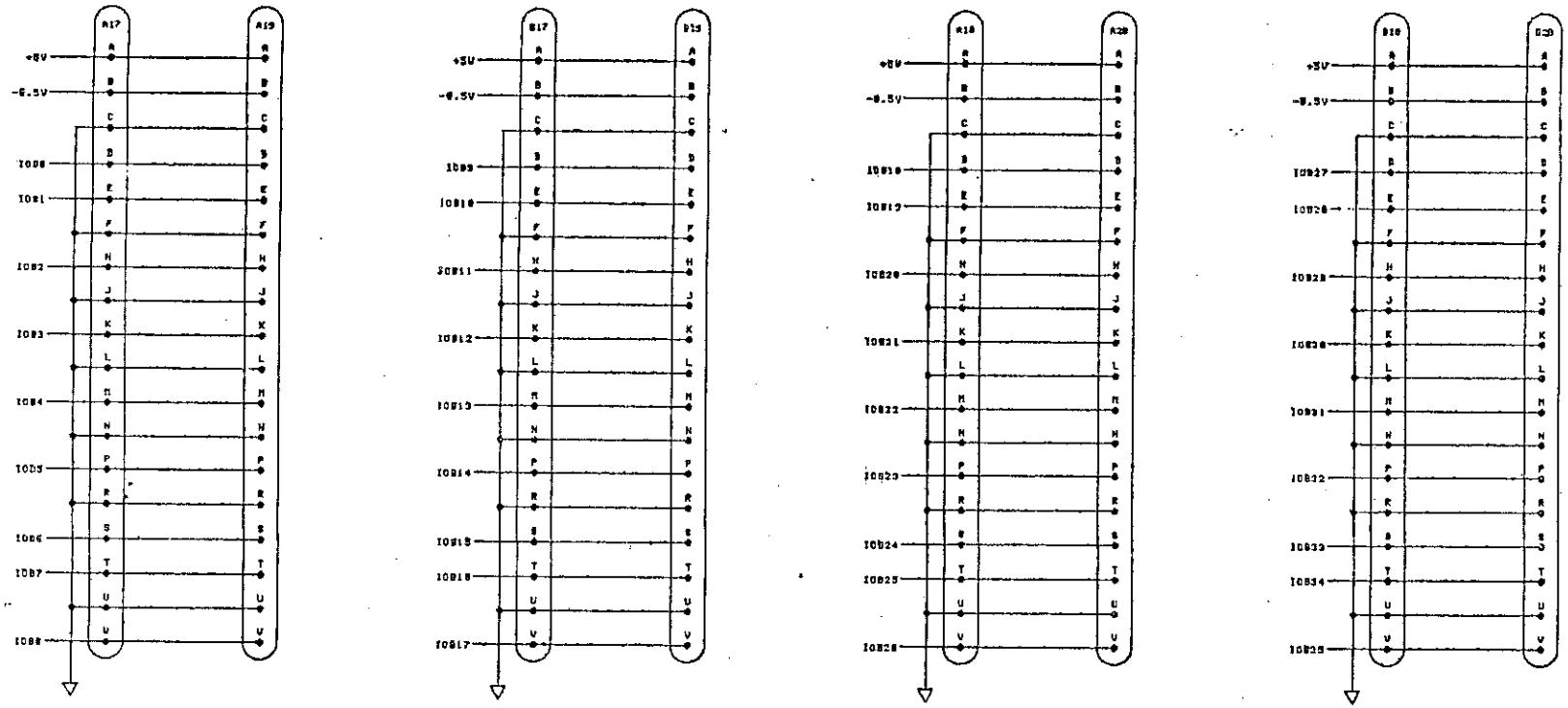
DxM;MDIAG1


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
MISC DIAGNOSTICS

DXM; MDIAG2


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
MISC DIAGNOSTICS

RLS. IOBCB1




SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

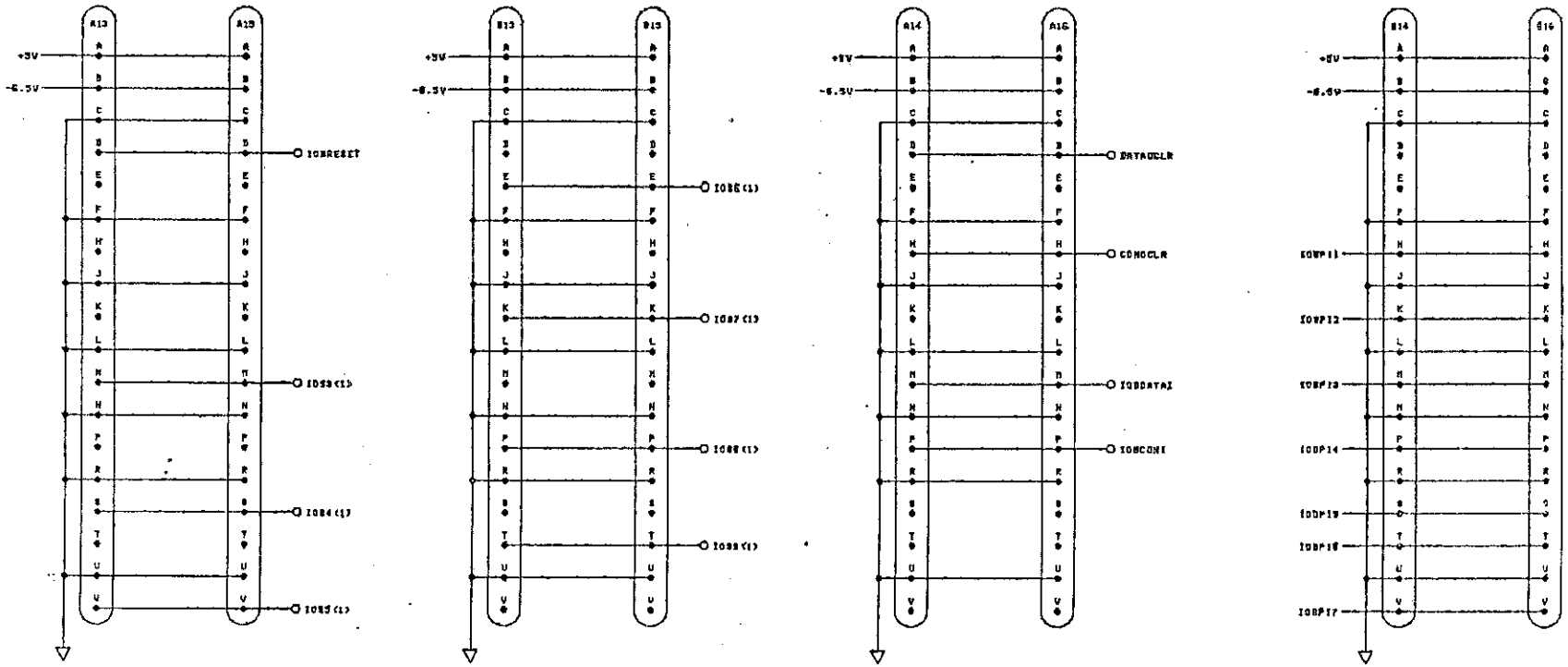
DIGITAL SYNTHESIZER

I/O BUS CABLE B1

| SIZE | CODE | DWG NO | REV |
|---|---|---|---|
| A | 1160 | | |

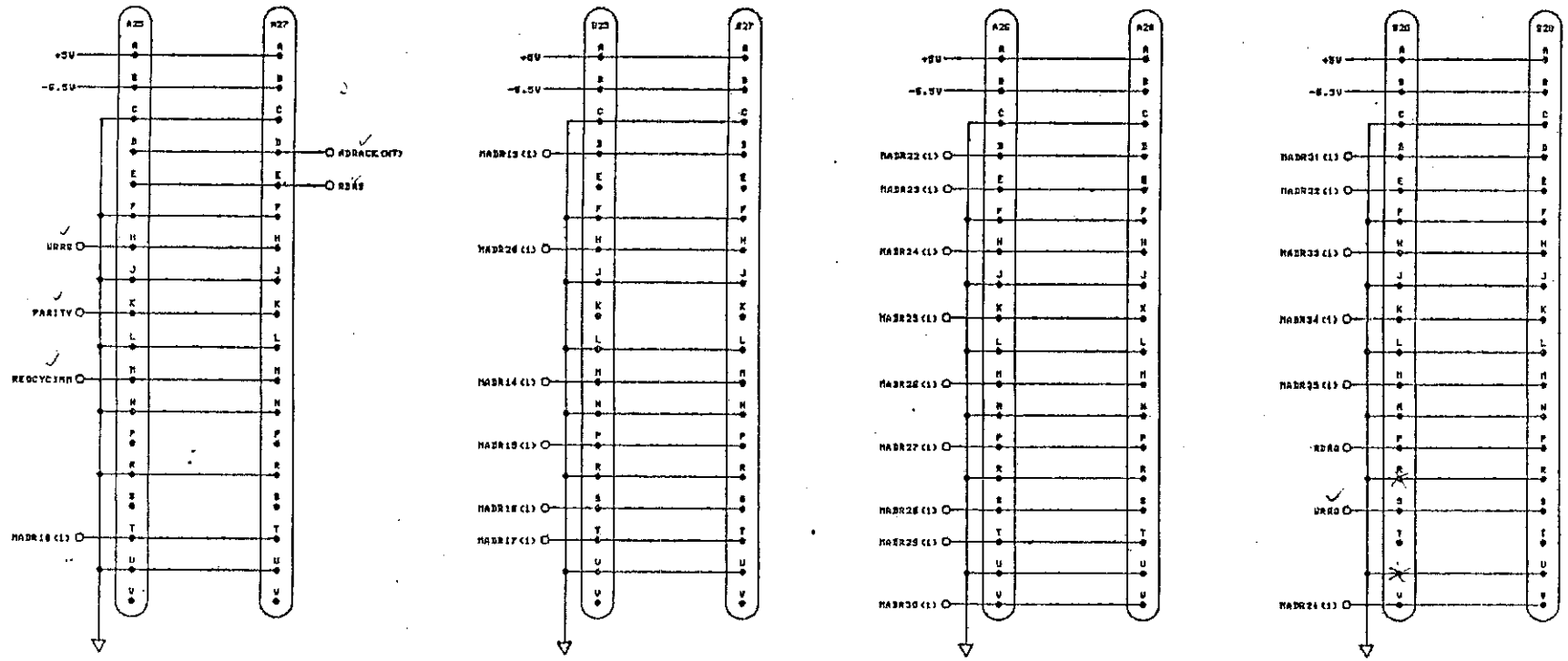
SCALE | SHEET | REV. | REFA. | PAGE

RLS; IOBCB2



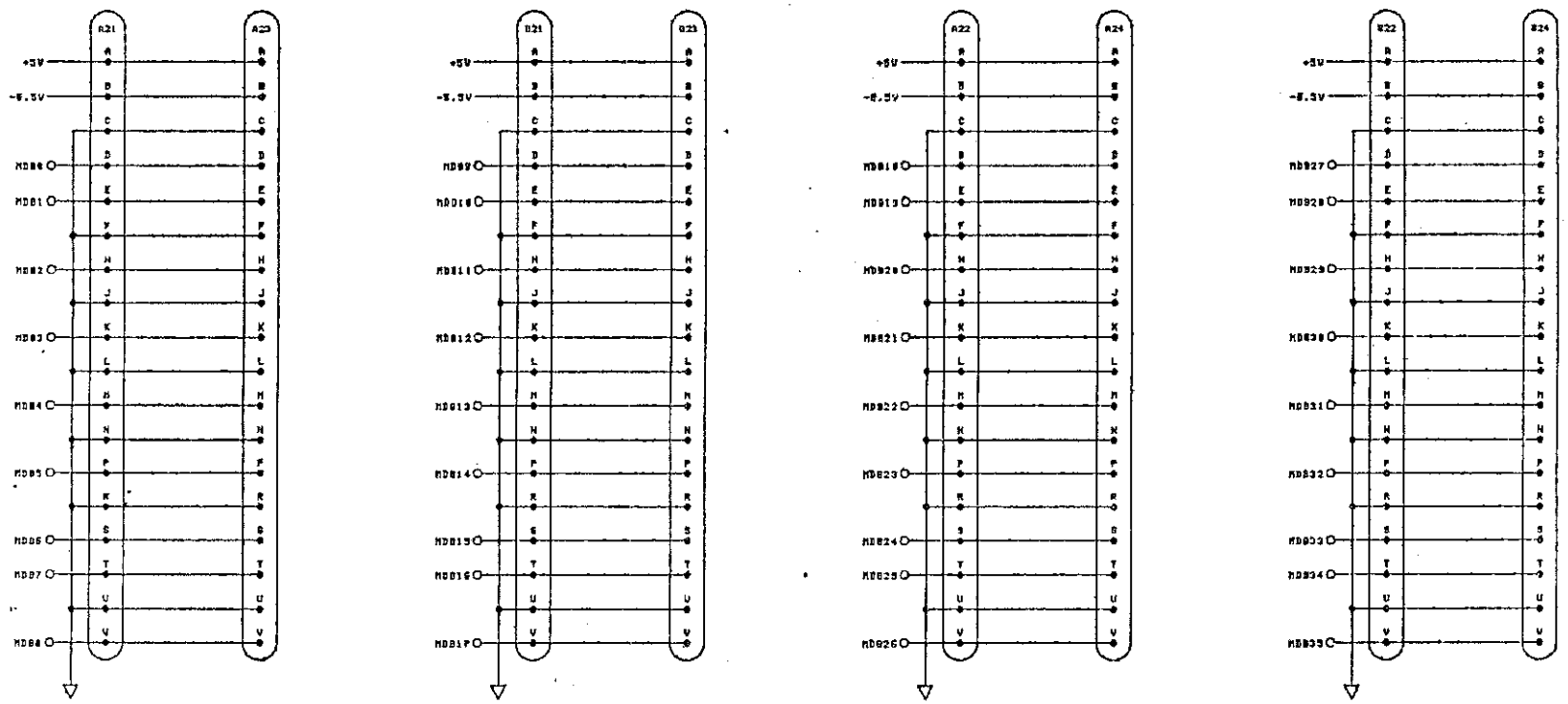

SYNTHES CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
I/O BUS CABLE B2

R25; MEMCB1



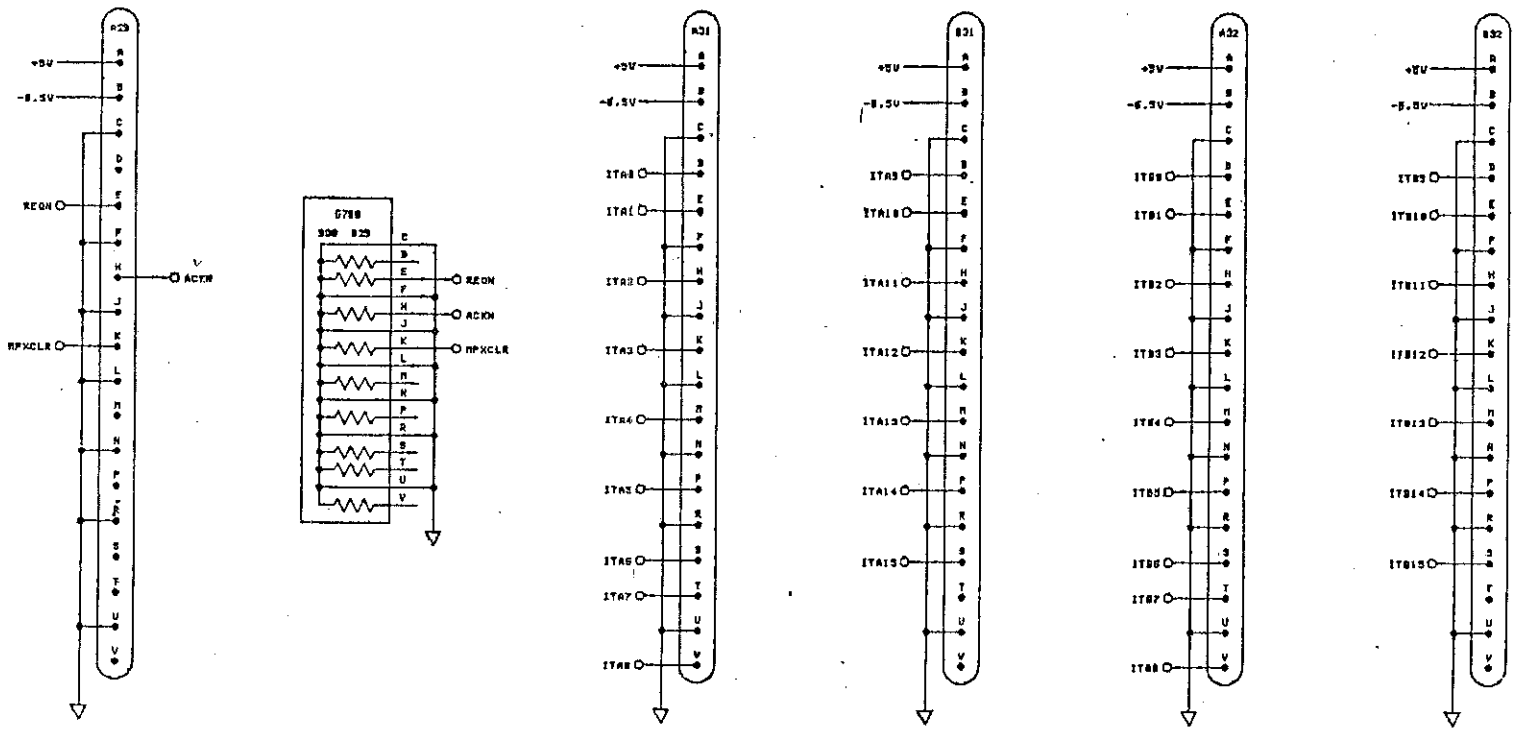

SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
MEMORY CARDS #1

RLS; MEMCB2




SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
DIGITAL SYNTHESIZER
MEMORY CABLE B2

RLS; PMTCB




SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

DIGITAL SYNTHESIZER

PORT MUX, TTL CABLES

MD820


MD821


MD822


MD823

```
        REPEAT NIPATV,<
                VALMAC REQ,\QX::        BLOCK 2
                QX=QX+1
                >>

EFINE   VALMAC (RR,QQ)
        <RR'QQ>

L:GENERAL PDP-10 CODE

AT 1000
```



```
        JRST @ERPNTX

NPNTX:  0
NPNTW:  0
NPNIT:  0
NPNTV:  BLOCK 2
RFLG:   0
RFLGA:  0
IE:     0

RPUK:   MOVE B,ERPNTW
        HRLOI D,377777
        MOVE E,D
        ASHC D,-105(B)
        POPJ P,

WP:     JUMPE D,TWP2
        MOVEI C,[ASCIZ /%D../]
        PUSHJ P,STYO
WP2:    MOVE A,E
        JRST OPT
LUUOH:  MOVEI P,PDL-1
        MOVEI C,[ASCIZ /**UUO %D/]
        MOVE D,40
        PUSHJ P,STYO
        JRST TE

NTTB:   INT1
        INT2
        INT3
        INT4
        INT5
        INT6
        INT7

NT1:    SKIPA L,[1]
NT2:    MOVEI L,2
        JRST INT8

NT3:    SKIPA L,[3]
NT4:    MOVEI L,4
        JRST INT8

NT5:    SKIPA L,[5]
NT6:    MOVEI L,6
        JRST INT8

NT7:    MOVEI L,7
NT8:    CAME L,PIE
        JRST INT8
        SETZM PIE
        PUSHJ P,DEVRST
        JRST 12,(MM)

NTB:    PUSHJ P,DEVRST
        MOVE MM,PIE
        SETZM PIE
        CONO PI,10000
        MOVEI P,PDL-1
        MOVEI C,[ASCIZ /**INTERRUPT ON CHAN %L, EXPECTING %MM/]
```

```
        AND D,VAL+1
        AND E,VAL
        PUSHJ P,SPACE
        PUSHJ P,TWP
        MOVEI C,[ASCIZ /: /]
        PUSHJ P,STYO
        PUSHJ P,ERPUK
        AND D,ERPNTV+1
        AND E,ERPNTV
        PUSHJ P,TWP
        PUSHJ P,CRLF
ERPNT9: DATAI DSW,A
        TLNE A,100000
        PUSHJ P,DEVRST
```



```
        PUSHJ P,STYO
        JRST TE

TESTN:: ANDI A1,177777
TESTN0: MOVEM A1,TESTNM
        SKIPN TESTMN
        JRST TESTN1
        CAML A1,TESTNN
        CAMLE A1,TESTNX
        JRST TESTN0
TESTN1: SKIPE ERFLGA
        AOS TESTER
        SETZM ERFLGA
        AOS TESTCT
        HRRM A1,PASSCT
        DATAO PI,PASSCT
        TROA A1,1
TESTNB: MOVEI A1,0
        POPJ P,

TESTCT: 0
TESTER: 0
^LPATA::        SETZM NPATS
        SETZM ERFLG
        JRST MKREST

PATN::  POP P,PATNR                     ;RETURN ADDR
        MOVE D,NPATS                    ;INDEX ON THIS PATTERN
        CAIL D,NIPATV-1
        JRST ERR1                       ;TOO MANY INDEPENDENT VARIABLES
        AOS NPATS
        MOVE A,PATN1
        MOVE E,(A)
        ADDI A,1
        MOVEM A,PATNQ(D)
        MOVEM A,PATNP(D)                ;PTR INTO PATTERN TABLE
        MOVE C,PATVP(D)
        MOVE B,PATNZ
        CAMLE B,E
        MOVE B,E
        CAMLE B,NTERMS
        MOVE B,NTERMS
        MOVEM B,PATLP(D)                ;# TERMS
        SETZM PATLC(D)                  ;CURRENT TERM
        PUSHJ P,PATNN
        JRST @PATNR

PATZ::  POP P,PATZR
        MOVEI D,0
PATZ1:  AOS PATNP(D)
        AOS A,PATNP(D)
        MOVE C,PATVP(D)
        PUSHJ P,PATNN
        AOS B,PATLC(D)
        CAMGE B,PATLP(D)
        JRST @PATNR
        MOVE A,PATNQ(D)
        MOVEM A,PATNP(D)
        SETZM PATLC(D)
        PUSHJ P,PATNN
```

SYSTEMS CONCEPTS

520 THIRD STREET SAN FRANCISCO, CALIFORNIA 94107

# SYSTEMS CONCEPTS

# ENGINEERING DRAWING

# CONVENTIONS

TELEPHONE: 415-442-1500 TWX: 910-372-6062

# SYSTEMS CONCEPTS LOGIC DRAWING CONVENTIONS

## Logic Blocks

A given logic block on a drawing contains the following information:

a) Abbreviated form of manufacturer's type number. E.G. 74175 is used for SN74175N, and 3000 for MC3000P.

b) Location. In a cabinet with several wire-wrap panels, for example, 2F11 means second panel from the top, section F, socket number 11. The area at the top of a section is divided into three 14-pin sockets, denoted 31, 32, and 33. Socket 31 is roughly over socket 1, 32 is in the middle, and 33 is roughly over socket 5. On a printed circuit board, designators such as U21 are used, keyed to a parts placement drawing for the board.

c) An asterisk if the block represents a 14-pin DIP which is in a 16-pin socket (or a section of such a DIP). In this case the DIP is always placed in the lower pins of the socket, such that each socket pin number is 1 greater than the corresponding DIP pin number.

d) Gates and inverters are indicated by the form of the logic block as drawn; more complicated functions will have a designation for each pin denoting its function. For instance, a J-K flip-flop might have terminals marked J, K, C, R, S, Q; C is the clock, R the Reset (clear), S the Set (preset), and Q the output.

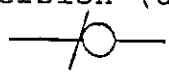
e) For each pin, the pin number. This is always the DIP pin number; if the chip is marked with an asterisk, add 1 to the given number to get the socket pin number.

## Mixed Logic

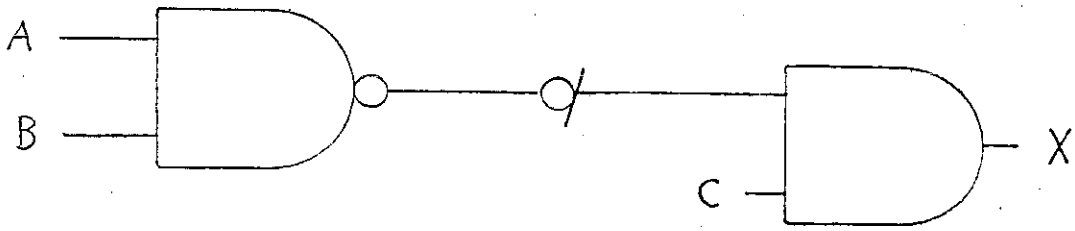
Mixed logic notation is used. An element is drawn showing the function intended by the designer, with inputs and outputs reversed in electrical polarity if needed. (See the example below.)

A signal asserted at a low voltage (ground in TTL) is shown with a nipple at each end of the wire. There may be a different signal with the same name, drawn without nipples, such as the other output of a flip-flop.

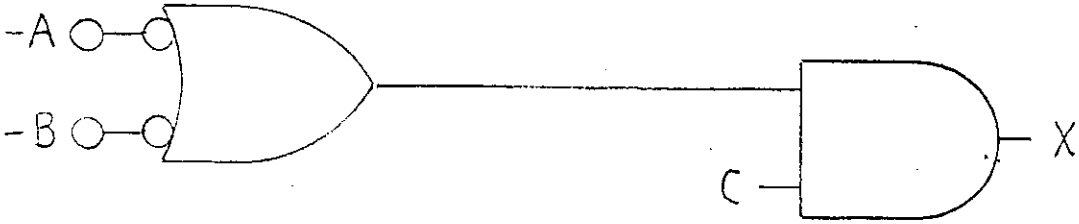
If an input is activated by the denial of a signal, a minus sign is used before the signal name. The signal -X without a nipple is electrically the same as X with a nipple; likewise, -X with a nipple is electrically the same as X without a nipple.

An inverter performs no logical function but only changes the polarity of assertion. Logical inversion (using an output to inhibit an input) is shown by a symbol like  .

$X = \overline{(A \wedge B)} \wedge C$ (all signals true high) would be drawn




$X = (\overline{A} \vee \overline{B}) \wedge C$ (all signals true high) would be drawn




Both drawings represent the same hardware and wiring.

## Clocking

Clock inputs are shown with the polarity of the pulse used, assuming a state change after the end of the pulse, i.e. activated by the trailing edge. This is consistent with the common practice of using the same signal both in clocking a device and in determining its mode or input data.

The gate input of a latch has its polarity shown such that assertion causes the latch to pass data from input to output.

## Ground, HI, Supply Voltages

Ground (0 volts) is drawn as a triangle pointed down or to the left.

HI (approximately +3 volts) is generated independently for each wire-wrap panel or printed circuit board. Unless otherwise indicated, the HI run connected to a device is the one specific to that panel or board.

Since ground and HI are fixed voltages rather than signals, the nipple of mixed logic is not drawn at the point where the HI label or ground triangle is attached to a wire or to a device.

Unless otherwise shown, ground and supply voltages are connected to each device according to the manufacturer's recommendation.

## Discrete Components

In printed circuit boards, discrete components are usually soldered in, and are given reference designators as follows:

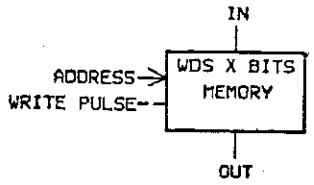
| | |
|---|---|
| Rn | Resistor or multiple-resistor package |
| Cn | Capacitor |
| Qn | Transistor |
| CRn | Diode |
| Tn | Transformer |
| Ln | Inductor |
| Jn | Jack |

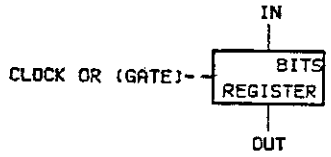
In wire-wrap panels, such components are usually mounted on 14- or 16-pin plugs which are then inserted into DIP sockets. The component is then designated by the DIP position in the panel and the socket pin numbers on which its terminals appear.

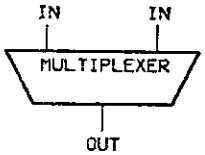
Unless otherwise indicated:

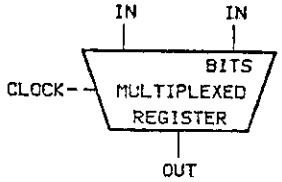
Resistors are 1/4 watt, 5%;
1% resistors are 1/8 watt;
Resistance values are in ohms;
Capacitance values are in microfarads.

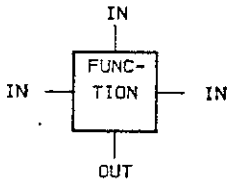
# SYSTEMS CONCEPTS BLOCK DIAGRAM CONVENTIONS




UNCONNECTED LINES

CONNECTED LINES

3-STATE BUS

SYSTEMS CONCEPTS 520 THIRD STREET SAN FRANCISCO, CALIFORNIA 94107

# SYSTEMS CONCEPTS

# DIGITAL SYNTHESIZER

# THEORY OF OPERATION



TELEPHONE: 415-442-1500 TWX: 910-372-6062

TABLE OF CONTENTS

# TAKE CARE

1. HAZARDOUS VOLTAGES EXIST WITHIN THE POWER CONTROL ENCLOSURE AND AT THE TERMINALS OF THE POWER SUPPLIES. TAKE APPROPRIATE PRECAUTIONS WHEN WORKING NEARBY.

2. NEVER INSERT OR REMOVE PRINTED-CIRCUIT CARDS OR PDP-10 CABLES WHEN POWER IS ON. FIRST MAKE SURE THAT THE FANS ARE FULLY STOPPED.

3. TO TURN POWER ON OR OFF, USE ONLY THE LOCAL-OFF-REMOTE SWITCH; DO NOT USE THE CIRCUIT BREAKER OR OTHER POWER LINE CONTROL.

4. DO NOT INSERT, REMOVE, OR CHANGE CARDS OR CABLES UNNECESSARILY.

5. WHEN INSERTING OR REMOVING A CARD, APPLY PRESSURE EQUALLY TO BOTH OF ITS EJECTORS. DO NOT USE EXCESSIVE FORCE.

6. FOR PREVENTIVE MAINTENANCE, PERIODICALLY MAKE SURE THAT COOLING AIR IS ENTERING THE CABINET AT THE BOTTOM AND THAT ALL SEVEN FANS ARE RUNNING WHEN POWER IS ON.

## Introduction

The Systems Concepts Digital Synthesizer is a large special-purpose digital processor that generates and modifies data streams that represent sounds and features of sounds like those found in music and speech. To accomplish its high computation rate, it operates several computational elements simultaneously and its data paths are extensively pipelined.

The synthesizer is a PDP-10 computer peripheral, with both I/O bus and direct memory access connections for a PDP-10 system. Its behavior and interface characteristics, as seen by the user, are described in the Systems Concepts Digital Synthesizer Programming Specification; an understanding of that document is assumed in the remainder of this manual. Familiarity is also assumed with the Systems Concepts Engineering Drawing Conventions.

An abbreviated notation is used in this manual for devices that are functionally equivalent to those in the 7400 series: for instance, 'H04 means 74H04 or equivalent. The terms .LT., .LE., .NE., .GE., .GT. mean respectively less than, less than or equal to, not equal to, greater than or equal to, greater than.

## Drawings

The engineering drawings for the Synthesizer include schematic drawings and a parts-placement drawing for each type of printed-circuit card, and logic drawings for the wire-wrap panels and cable connection rack. For each type of card, the card drawing gives a set of generic names for the signals on any card of the type; and an accompanying table relates each generic name to the specific name of each signal on each card of the type. In the generic names, the term (M) is used for the base bit number in a four-bit slice; for example, the generic name FG(M) would correspond to FG0 on the most significant card of the type, to FG4 on the next card, and so on.

The following drawing numbers are used for the Synthesizer:

1110, 1110M, 1111.0, 1111M, 1112.0, 1112M, 1113.0, 1113M, 1114.0, 1114M, 1115.0, 1115M, 1116.0, 1116M, 1117.0, 1117M, 1118.0, 1118M, 1119.0, 1119M, 1120.0, 1120M, 1121.0, 1121M, 1122.0, 1122M, 1123.0, 1123M; 1160, 1161, 1162, 1163, 1164; 1171, 1172, 1173, 1175, 1176, 1177, 1179, 1180, 1181, 1182, 1184, 1185, 1187, 1188, 1190, 1191, 1193, 1194, 1195, 1197.

In this manual, a reference to drawing number 1191 (for example) is abbreviated to #1191.

## Packaging

The Synthesizer is housed in a single free-standing cabinet. It contains the following, from top to bottom: five power supplies; three wire-wrap panels; two printed-circuit-card chassis, with backpanels designated YBACK (upper) and ZBACK (lower); seven cooling fans; a cable connection rack; and the power control enclosure.

The five power supplies and their uses are as follows:

| Model No. | Voltage(s) | Rated Current | Use |
|---|---|---|---|
| LGS-EE-5 | +5 V | 110.0 A | YBACK, ZBACK |
| LXS-D-5 | +5 V | 27.5 A | Wire-wrap Panels |
| LXS-A-5 | -5 V | 4.0 A | Delay Memory, DEC Bus Interfaces |
| LXS-A-12 | +12 V | 2.7 A | Delay Memory |
| LXD-C-152 | +,-15 V | 2.5 A each | Analog Outputs |

Each supply has an overvoltage protector.

The three wire-wrap panels hold ICs (integrated circuits), and a few discrete components, that comprise the once-only logic of the Synthesizer. The panels are numbered 1, 2, and 3, from the top.

The card chassis hold the printed-circuit cards of several types that are replicated in the system. A typical card has the hardware dealing with successive stages of a four-bit-wide slice of a data path. A data path wider than four bits is processed by a group of cards of the same type; for instance, five cards will be grouped together to process a 20-bit-wide section of the Synthesizer. Viewing the backpanels from their wiring side, card slots are numbered from left to right: Y1-Y39 on YBACK, Z1-Z39 on ZBACK (but some slots are not used). In a group of cards of the same type, the leftmost one processes the most significant bits of the data. The cards in the system are tabulated below.

| Card Name | Dwg # | Short Name | Slots |
|---|---|---|---|
| Dual Analog Output | 1110 | ALOG | Y1-4, Z1-4 |
| Miscellaneous-A | 1118 | MISCA | Y8-11 |
| Sum Memory | 1117 | SUM | Y12-16 |
| Miscellaneous-B | 1119 | MISCB | Y17-19 |
| Modifier-A | 1114 | FILTA | Y20-24 |
| Modifier-B | 1115 | FILTB | Y25-29 |
| Delay Memory Data | 1120 | DMD | Y30-34 |
| Delay Memory | 1121 | DMEM | Y35-38 |
| Generator-C | 1113 | GENC | Z6-8 |
| Multiplier | 1116 | MULT | Z9-13, Z22-26 |
| Generator-B | 1112 | GENB | Z14-16 |
| Generator-A | 1111 | GENA | Z17-21 |
| 36-bit Interface | 1123 | TENI | Z27-35 |
| Generic Interface | 1122 | INTF | Z36-39 |

On a card, each IC is designated with a U number. Viewing a card in the orientation in which it is plugged in, U1 is at the top next to the edge connector and U2 is beneath it. Counting continues first downward by row, then outward by column. Positions without signal wiring are not counted.

The cable connection rack (#1160 through #1164) provides sockets for the I/O bus, memory bus, memory port multiplexer cable, and the outputs of the TTL output registers. The upper connector row is designated A and the lower B; slots are numbered 13 through 32, from left to right.

In the power control enclosure are the power control board, Local-Off-Remote switch, circuit breaker, and relays that control sequencing of the power supplies.

## Controls and Indicators

Since the Synthesizer is designed for checkout by computer, there are very few manual controls and indicators. The only indicator is a red LED, on the power control board, which is lit when AC power is applied to that board. It is not visible unless the louvered cover of the power control enclosure has been removed. However, the sound of the fans is a clear indication that the power supplies are also on. On the power control panel are the main circuit breaker and, to turn the Synthesizer on and off, the Local-Off-Remote switch.

In location 3A33 is a package containing seven on-off switches. Switches 3-7 give the base device address of the Synthesizer on the I/O bus, and correspond respectively to IOS3-7 in the PDP-10 I/O structure. Switch 1 can be used to prevent the Synthesizer from writing into PDP-10 memory.

## Signal Names

Data signals generally have names composed as follows:

a) One letter indicating the general area involved:

- A analog
- C clocking
- D diagnostics
- F modifiers ("filters")
- G generators
- I generic interface
- P phases of clock
- R delay units ("reverb")
- S sum memory
- T PDP-10 interface ("ten")
- U unused, provided for possible future additions

b) One or two letters arbitrarily chosen to distinguish busses in a general area;

c) A decimal number for bit position (0=most significant) in a bus, or for a decoded value of a field (0=all bits off).

A control signal is usually named by appending, to the name of the bus it controls, a letter to designate the signal's function. For instance, the clock to the FE register (bits FE0-19) is called FEC. Some of the more common functions are: C -- clock; E -- enable; G -- gate (of a latch); R -- reset; S -- select (ALU mode select or multiplexer input select). A control signal with several functions may be named instead for its derivation.

Names ending in -A, -B, etc., but otherwise alike, denote signals that are logically equivalent but physically distinct, as for loading purposes. Names on the drawings ending in -1, -2, etc. represent signals which are logically equivalent but generated on different printed circuit boards. In this manual, however, a notation such as BUS0-19 means the 20 bits BUS0 through BUS19. A number within a signal name, surrounded by letters, either denotes the time state of a quantity used at different stages of a pipeline or denotes a quadrant of sum memory.

XHI and XGND are forms of HI and GND brought onto printed-circuit cards from the backpanels through signal pins. The versions of HI for the wire-wrap panels and for XHI (named YHI for YBACK and ZHI for ZBACK) are generated by a resistor package shown on #1171.

## Clocking

Clock generation for the Synthesizer, shown on #1171, has a 30,769,230-hertz crystal, whose output is divided by 6. A 10-pF capacitor in series with the crystal trims the oscillator to the specified frequency. Outputs of the frequency divider (three Schottky J-K flip-flops) go through delay lines and AND gates to form the basic clock pulses shown in Fig. 1. Each 195-nsec tick (also called a time state) has three equally-spaced clock pulses termed phases A, B, and C; there is also a pulse roughly halfway through the time state (between phases A and B) called phase H. The corresponding signals are CA, CB, CC, and CH. Each pulse is nominally 45 nsec wide. The trailing edge of phase C marks the end of each time state. #1171 also shows the formation of several special-purpose clocks which occur more than once per time state: SAC and CU, which occur on phases A, B, and C; and CHC, occurring on phases H and C.

The basic clock pulses are buffered for distribution throughout the Synthesizer by gating shown on #1172. The signals for the various phases take on the names PHA, PHB, PHC, and PHH. These are conditioned by the clock enable flip-flops, CEAB (for phases A, H, and B) and CEC (for phase C). Ungated clocks designated PHAU, PHBU, and PHCU are also created for the PDP-10 interface and delay memory control, which must run even when other activity in the Synthesizer is stopped.

The clock-enable state depends on the CRUN flip-flop, which is direct-set by the clock-start CONO-A; it is direct-cleared by master reset, the clock-stop CONO-A, or performance of the clock-stop command; and it is clocked off by occurrence of a 10AAA cause enabled by CONO-B. CRUN is ORed with the clock-one-tick CONO-A and the result is ANDed with terms indicating that the clock is not held for read or write data direct memory access (underrun conditions), to produce CRUNA; this is clocked into CEAB, which in turn is clocked into CEC.

Various clock counting functions are shown on #1173. The time state pipeline, on the MISCA cards, appears on sheet 2 of #1118. Fig. 2 shows the relationship of the principal signals involved. CTK0-9 counts the ticks of a pass. Its count is compared against two registers loaded by commands: CTP0-9, denoting total processing ticks, and CTT0-9, denoting total ticks per pass. The EPAS flip-flop is on for the last tick of a pass: it is direct-set by the reset-tick-counter CONO-A; held direct-cleared in the "all ticks update" state; clocked on when CTK0-9 equals CTT0-9; and clocked off at the end of the tick when it is on. Among its effects, EPAS conditions the CTK0-9 counter to parallel-enter zero at the end of the tick.

Generator and modifier calculations each require 9 steps of pipelining; the first is tick A for a given generator or modifier; then follow its ticks 0 through 7. Tick A is preparatory; during it various sum memory addresses are determined. Numeric processing does not start until tick 0. At any point in the generator data paths, data for successive generators is processed on successive ticks. In the modifier data paths, there are two ticks in a row for each modifier. For instance, if CP0B0-7 equals 2, it is tick 2 for for generator 0, tick 1 for generator 1, tick 0 for generator 2, tick A for generator 3; tick 2 for modifier 0, and tick 0 for modifier 1. The time-state pipeline consists of the CPAB0-7 counter and the CPnB0-7 shift registers. The CPnBm busses are used to address RAMs holding data for each generator or modifier, where n denotes the tick during which the RAM is referenced. CPAB0-7 during the processing ticks of a pass counts the same as CTK2-9; CP0B0-7 during processing tick n has the value CPAB0-7 had in tick n-1; CP1B0-7 has that value in tick n+1; and so on. During the interface ticks (also called update ticks) all the CPnBm terms are parallel-loaded with the generator number or modifier number of the next command to be performed (the interface address, IA0-7). CTRA is turned on at the beginning of a pass, when CTK0-9 is reset; it is turned off when CTK0-9 equals CTP0-9, to flag the end of the processing ticks. CTRA is on during a valid tick A for some generator and modifier. Similarly, CTR0 through CTR7 are on during valid processing ticks 0 through 7. CTRA and CTR0-7 are held off in "all ticks update" mode. Flip-flop ITR is on when "real" (i.e. processing) ticks are in progress anywhere in the pipeline; it is off during update ticks. The signal OT, meaning odd tick, is on for tick A of a pass (CPABn = 0), off for tick 0 (CP0Bn = 0), and so on. Similarly, OP means odd pass; it changes state at the same time that CTRA comes on.

Generator Data Paths

Fig. 3 is a block diagram of the generator data paths. Data processing for a generator, tick by tick, proceeds as follows.

Tick 0 -- Oscillator side: On phase A, the 'LS195A registers holding GOA0-7 on PC cards clock in CP0B0-7. This addresses the 256-bit RAMs holding GO0-19 (GENA) and GJI0-27 (MISCB, GENA) (called GJ in the Programming Specification). On phase B, the fm term from sum memory is clocked into 93H72s forming GRA0-19 (SUM). On phase C, GJI0-27 is latched in the 'LS157 multiplexers forming GJ0-27 (MISCB, GENA).

Tick 1 -- Oscillator side: During phase A, GO0-19 with sign extended is added to GJ0-27 by a fast adder, using 'S181 ALUs and 'S182 carry generators, to form GAZ0-27 (MISCB, GENA). The phase A clock writes this sum into GJI. At the same time, the ripple adder GA0-19 (GENA), using '283s, forms the sum of GJ0-19 and GRA0-19, which is clocked by phase A into the 'LS175s GWA0-19 (GENA). The 256-bit RAMs comprising GK0-19 (GENA) and GN0-10 (GENC) are addressed by CP1B0-7, buffered by 'LS04 inverters. On phase B, GK0-15 is latched into the 'LS157 multiplexers GXA0-19 (GENA). On phase C, the high-speed adder GB0-19 (GENA) (similar in structure to GAZ) develops the sum of GWA0-19 and GXA0-19. (Due to the polarities of the signals involved, the adder is actually configured to subtract the ones' complement of GXA, with a borrow, from GWA.) The phase C clock pulse writes GB0-19 into GK0-19, and clocks the 'LS298 register GXB0-12 (GENB, #1179) from either GXA0-12 or GWA0-12, according to the generator mode (the GXB12 selection is done by an 'H51 gate). Also on phase C, GN0-10 is clocked into the '175s GXC0-10 (GENC).

Tick 2 -- Oscillator side: GXB0-12 is combined with GXC0-10 in a modified Wallace tree (MULT, #1179) to form the low-order 13 bits of the expression GXB * (2 * GXC + 1) + GXC. (This has the effect of multiplying GXB by 2 * GXC + 1 if GXB and the product are both assumed to have an implied bit 13 equal to 1.) The 'LS298 register GX0-12 (GENB, #1179) is clocked on phase C from either the product or GXB0-12, depending on the generator mode. GWB0-12 ('LS175s on GENB) is clocked from GXB0-12 on phase C also.

Tick 3 -- Oscillator side: The sine of GX and the cosecant of GWB are looked up in ROM. The assumed low-order 1-bit in each case results in two simplifications: (1) GX0 and GWB0 need not be looked up, but are saved to govern whether the looked-up value will be negated; (2) GX1 and GWB1 need not be looked up, but merely cause ones'-complementing of GX2-12 or GWB2-12 respectively, if set. The ones'-complementing of GX2-12 is done by 'LS86s (GENB). For GWB2-12, 'H87s (GENB) are used in order to substitute all-ones (whose cosecant is approximately 1), except in sum-of-cosines mode. The sines and cosecants are stored in floating-point form in 512x4 PROMs on the MISCA cards. The low two bits of each address are decoded to select one of four banks of PROMs; the remaining nine bits address the PROMs through 'H04 buffers. The PROM outputs are clocked into registers on phase C as follows:

| | Sine Exponent, Fraction | Cosecant Exponent, Fraction |
|---|---|---|
| PROM Output | GGE0-3, GGF0-11 | GFE0-3, GFF0-11 |
| Register | GYC0-3, GYD0-11 | GYA0-3, GYB0-11 |

Exponent registers are on GENC, and fraction registers on GENB. Also on phase C, GWB0-11 is clocked into GWC0-11; and the RAM GM0-3 (MISCB, addressed by CP3B0-7 through 'LS04 inverters) is clocked into GYE0-3 (GENC).

Tick 3 -- Envelope side: Phase A clocks CP3B0-7 into the 'LS195As serving as address registers for the RAMs GP0-19 (GENA) and GQ0-23 (MISCB, GENA). During phase C, GQ0-23 is latched into the 'LS157s comprising GVA0-23 (MISCB, GENA); the end of phase C clocks GVA0-13 into GVB0-13 (#1179, MISCB).

Tick 4 -- Oscillator side: GYB0-11 is multiplied by GYD0-11 and the high-order 12 bits of the product are clocked on phase C into the 93H72s comprising GYP0-3 and GY4-11 (GENB) (GYQ3, clocked at the same time, is a late output from the Wallace tree which is added to GYP0-3 during tick 5 to correct the product). The exponents GYA0-3 and GYC0-3 are added together with scale factor GYE0-3 and the results clocked on phase C into the 'LS175 termed GYG0-3 (#1179). GWC0-11 is clocked on phase C into GWD0-11 (GENB).

Tick 4 -- Envelope side: GVB0-13 is treated as a negative exponent of 2, with a binary point between GVB3 and GBV4. The field GVB4-13 addresses the PROM GH0-11 (MISCA), the bits GVB4-12 addressing the PROMs directly and bit GVB13 in true and ones'-complement forms enabling one or the other bank of PROMs. Then GH0-11 is run through 'LS153 multiplexers (GENC), configured to shift right 0, 1, 2, or 3 places according to the value of GVB2-3. (If GVB0-1 = 11, the 'LS153s are disabled, producing zeros.) The result (GIM0-11) goes into 9309 multiplexers (GENC) that can shift right 0, 4, or 8 places (according to GVB0-1) or substitute GVB0-11 if in linear mode. This result, GIN0-11, is available in both polarities; on phase C, one or the other polarity is clocked into the GT register ('LS298s on GENC) according to the envelope mode.

Tick 5 -- Oscillator side: GY0-11, the fraction part of the floating point product, is shifted right 0 to 15 places by two banks of 'LS153s (GENB) according to the value of GYG0-3, the exponent part; the result is GIB0-11 (not named in the schematic drawing). A term GWE0-11 is derived by 'LS86s and '283s (GENB) as follows: GWE0-10 is GWD1-11, two's-complemented if GWD0 is 1; GWE11 is 0. Then GU0-11, the result of the envelope side (except for the sign, which is handled by control logic), is selected by 'LS153 multiplexers (GENB). It is either GIB0-11 (sine or sum-of-cosines mode), GWE0-11 (sawtooth mode), an overflow bit from the GA and GB adders (pulse-train mode), or 4000 octal (square-wave mode). On phase C this result is clocked into GZA0-11 (93H72s on GENB).

Tick 5 -- Envelope side: The RAM GL0-11 (GENC), addressed by CP5B0-7 through 'LS04 inverters, is added to GT0-11. A carry is injected in the low-order position if the ones'-complemented version of GIN was taken in tick 4, thereby accomplishing a two's-complement negate. The sum GE0-11 (GENC) is formed by a ripple-carry adder of '283s; it is clocked on phase C into GZB0-11 (93H72s on GENC).

Tick 6 -- The unsigned quantities GZA0-11 and GZB0-11 are multiplied (MULT, #1179) and the high-order 18 bits of the product are clocked on phase C into GZ0-17 (93H72s on #1179); again, one late bit, GZC3, is clocked at the same time to be added in later.

Tick 7 -- GZC3 is added to GZ0-3 by a '283 to form the correct high-order product bits GZS0-3. During clock phase CCB (see Fig. 1) the contents of the sum memory location to be augmented are latched into the 'LS157s GRB0-19 (SUM). On the FILTA cards, GZS0-3 and GZ4-11 are ones'-complemented by 'LS86s if the result should be negative, then added (sign extended) by '283s to GRB0-19, with a carry in if necessary to complete a two's complement. The sum, GF0-19, is returned to sum memory where it is written on phase C.

Modifications in certain generator run modes: If the oscillator side is not running, write pulses are not given to the GJI and GK memories. If the envelope side is not running, write pulses are not given to the GQ memory. If a generator is not to add to sum memory, the sum memory write pulse is not given. If a generator is reading data from PDP-10 memory, the data read appear in GRB instead of the contents of the sum memory word being augmented. If a generator is feeding a DAC, GRA0-19 is clocked into AP0-19 ('175s on SUM) at the end of phase B, tick 1; and AP0-13 is clocked into the proper DAC hold register at the end of phase A, tick 2. If a generator is writing data into PDP-10 memory, GRA0-19 is clocked into IWB0-19 (SUM) on phase B of tick 1.

Command execution: During interface ticks, all CPnB0-7 hold the generator number from the command, clocked in on the previous phase C. Address registers (such as for GO and GQ) are clocked from CPnB0-7 on phase A. Data to be written comes direct from the generic interface into memories GO, GN, GM, GP, and GL. To write GJI or GQ, data from the interface is introduced by 'LS257 3-state multiplexers instead of the GO or GP RAMs (which are disabled), and the GAZ or GC adder is put in the mode where it passes the "A" input through to the output. To write GK: GWA is held reset during interface ticks; during phase B, the GK memory is disabled. 'LS257s are enabled to place interface data on the GK lines; GK is latched into GXA at the end of phase B. All commands cause memories to be written on phase C.

## Generator Control

The generator run mode, GRMD0-3, is stored in RAMs on a MISCB card, addressed by GRMDA0-7 on #1175. GRMDAn (93H72s) is clocked from CPABn on phase H and from CP4Bn on phase C. The mode for a generator is read on the second half of the generator's tick A, and clocked on phase C into GRMD0B0-3 ('S175 on #1175). The mode bits go through a pipeline of 'LS174s through GRMD5Bn; on phase H of tick 5 these bits are written back into GRMDn. Along the way the mode may have been altered by IRP (clear all pause bits), IRW (clear all wait bits), GCOD (envelop overflow), or GT (trigger from previous generator). The GRMDn write pulse, GRMDW, also occurs by command (IIM50).

The other mode bits, corresponding to bits I7-12 of the command data, are shown on #1177. GUS0-1 select the waveform; GXS selects the output of GX; GXBS selects the signal input to GX; GTS selects whether the envelope is added or subtracted from the asymptote; and GINVBE chooses between linear and exponential envelope modes. Each RAM is addressed by the time state in which it is used.

Straightforward gating (#1176, #1177), based on the mode bits and the time state, creates the enables, write pulses, clocks, and selects required for the processing described in "Generator Data Paths" above.

## Modifier Data Paths

Fig. 4 is a block diagram of data paths for the modifiers. The constituents, and the boards they appear on, are as follows:

| | |
|---|---|
| FL0-19: | 256-word RAM holding the L0 and L1 terms, written from the FLI bus (FILTA) |
| FE0-19: | 93H72s, clocked from FL0-19 (FILTA) |
| FEE0-19: | 93H72s, clocked from FL0-19 (FILTA) |
| FN0-19: | 'LS670s, written from FE0-19; addressing accomplishes a 3-stage delay advanced every other tick (FILTB) |
| FG0-19: | 'S175s, clocked from sum memory (FILTA) |
| FV0-19: | 93H72s, clocked from sum memory (FILTA) |
| FW0-19: | 'LS157s, latched from sum memory (FILTA) |
| FUI0-29: | 256-word RAM holding the M0 and M1 terms, written from FA adder; 3-stated with 'LS258 multiplexers from generic interface (MISCB, FILTA) |
| FU0-29: | 'LS157s, latched from FUI0-29 (MISCB, FILTA) |
| FA0-29: | fast adder ('S181s and 'S182s); adds FU0-29 and FG0-19, sign extended (MISCB, FILTA) |
| FQI0-19: | 'LS175s clocked from FUI0-19 (FILTA) |
| FP0-19: | 'S257s 3-stated with 'S258s (not named on schematic); inputs are: FEE0-19; R0-19 (from delay memory); FG0-19; sum memory (FILTA) |
| FQ0-19 | 9309 multiplexers (not named on schematic); inputs are: FQI0-19; FV0-19; 0, -FV0, FV1-18; 0 (FILTA) |
| FXA0-19: | 93H72s clocked from FP0-19 (FILTA) |
| FXB0-19: | 93H72s clocked from FQ0-19 (FILTA) |

FXP, FXH, FXJ: intermediate stages of FX multiplier (see below) (FILTB)

FX0-19: 93H72s, final product of FXA and FXB (FILTB)

FK0-19: 'LS175s clocked from FX0-19 (FILTB)

FT0-19: 'LS157s, can provide: 0, FK0-19, or FN0-19 (FILTB)

FB0-19: ripple-carry adder of '283s; adds FT0-19 and FX0-19 (FILTB)

FJ0-19: 'LS175s clocked from FB0-19 (FILTB)

comparators indicating FJ .GT. FK, FJ = FK, FJ .LT. FK (FILTB, #1182)

FR0-19: 'LS258s 3-stated together; inputs are: FK0-19; FJ0-19; FF0-19; FD0-19 (FILTB)

FC0-19: fast adder ('S181s, 'S182) adding FR0-19 and FW0-19 (FILTB)

FY0-19: 'LS157s; inputs are: FJ0-19; FC0-19 (FILTB)

FD0-19: 'LS175s clocked from FY0-19 (FILTB)

FZ0-19: 9309s; inputs are: FJ0-19; FC0-19; FK0-19; FW0-19 (FILTB)

FF0-19: 'LS175s clocked from FZ0-19 (FILTB)

FLI0-19: 'LS257s 3-stated together; inputs are: FD0-19, FF0-19, I0-19 (FILTB); FE0-19, R0-19 (from delay memory) (FILTA)

The processing steps are listed in order below for each of the modifier modes. The notation "1A", for instance, means phase A of a modifier's tick 1. The FX multiplier is discussed in a separate section below.

Two Poles, Two Zeros (possibly M0 or M1 variable)

0H: L1 to FE, FEE

0B: M1 to FU if M1 variable

0C: FEE to FXA; L0 to FEE; M1 to FXB; sum(MRM) to FG

1H: FE to L0

1B: M0 to FU if M0 variable

1C: FEE to FXA; M0 to FXB; FA to M0 if M0 variable, or FA to M1 if M1 variable

4C: FX (L1 * M1) to FK

5C: FX (L0 * M0) + FK to FJ

6A: sum(MIN) to FW

6C: FJ + FW to FD; FW to FF

7B: sum(MSUM) to FW

7C: FD to L1 if two poles, or FF to L1 if two zeros; FW + FD to sum(MSUM)

Mixing, Integer Mixing

0B: if MIN is in modifier-this-pass quadrant, sum(MIN) to FG

0C: sum(MRM) to FXA; M1 to FXB

1A: if MIN is not in modifier-this-pass quadrant, sum(MIN) to FG

1C: FG to FXA; M0 to FXB

4C: FX (B * M1) to FK

5C: FX (A * M0) + FK to FJ

6C: FJ to FD

7B: sum(MSUM) to FW

7C: FW + FD to sum(MSUM)

Amplitude Modulation, Four-Quadrant Multiplication

0H: L1 to FEE

0B: if MIN is in modifier-this-pass quadrant, sum(MIN) to FG

0C: FEE to FXA; M1 to FXB; sum(MRM) to FV

1A: if MIN is not in modifier-this-pass quadrant, sum(MIN) to FG

1C: FG to FXA; if four-quadrant multiplication, FV to FXB; if amplitude modulation, 0 to FXB0, -FV0 to FXB1, FV1-18 to FXB2-19

4C: FX (L1 * M1) to FK

5C: FX (B * A) to FJ

6C: FK to FF; FJ to FD

7B: sum(MSUM) to FW

7C: FD to L1; FW + FF to sum(MSUM)

Minimum, Maximum

0B: if MIN is in modifier-this-pass quadrant, sum(MIN) to FG

0C: sum(MRM) to FXA; M1 to FXB

1A: if MIN is not in modifier-this-pass quadrant, sum(MIN) to FG

1C: FG to FXA; M0 to FXB

4C: FX (B * M1) to FK

5C: FX (A * M0) to FJ

6C: FK to FF; FJ to FD

7B: sum(MSUM) to FW

7C: FD or FF (depending on mode and comparison FJ:FK) + FW to sum(MSUM)

Zero-Crossing Pulser

0H: L1 to FEE

0C: FEE to FXA; M1 to FXB; sum(MRM) to FG

1C: FG to FXA;M0 to FXB

4C: FX (L1 * M1) to FK

5C: FX (B * M0) to FJ

6C: all-ones to FD; FJ to FF

7B: sum(MSUM) to FW

7C: FF to L1; FW + FD + (0 if FK is not 0 and if either FJ is 0 or FK * FJ is negative; else 1) to sum(MSUM)

Invoke Delay Unit

0H: L0 to FE

0C: R (delay memory) to L0, FXA; FE to FN; M1 to FXB

1H: L1 to FEE

1C: FEE to FXA; M0 to FXB

4C: FX (R * M1) to FK

5C: FX (L1 * M0) + FN (L0) to FJ

6A: sum(MIN) to FW

6C: FW + FK to FF, delay memory; FJ to FD

7B: sum(MSUM) to FW

7C: FF to L1; FW + FD to sum(MSUM)

Latch

0H: L1 to FE

0C: FE to FN; sum(MRM) to FXA; M1 to FXB

1C: 0 to FXB

4C: FX (B * M1) to FK

5C: FX (0) + FN (L1) to FJ

6A: sum(MIN) to FW

6C: FW to FF; FJ to FD

7B: sum(MSUM) to FW

7C: FF to L1 if FK is not 0; FW + FD to sum(MSUM)

Uniform Noise, Triggered Uniform Noise

```
0H:  L0 to FE
0C:  FE to FN; sum(MRM) to FXA; M1 to FXB
1H:  L1 to FEE
1C:  FEE to FXA; M0 to FXB
4C:  FX (B * M1) to FK
5C:  FX (L1 * M0) + FN (L0) to FJ
6C:  FJ to FD
7B:  sum(MSUM) to FW
7C:  FD to L1 if FK is not 0, or not in triggered mode;
       FW + FD to sum(MSUM)
```

One Pole

```
0H:  L1 to FEE
0C:  FEE to FXA; L0 to FEE; M1 to FXB; sum(MRM) to FV
1C:  FEE to FXA; FV to FXB
4C:  FX (L1 * M1) to FK
5C:  FX (L0 * B) + FK to FJ
6C:  FJ to FD
7B:  sum(MSUM) to FW
7C:  FW + FD to sum(MSUM)
```

One Zero

```
0H:  L1 to FE, FEE
0C:  FEE to FXA; L0 to FEE; M1 to FXB
1H:  FE to L0
1C:  FEE to FXA; M0 to FXB
4C:  FX (L1 * M1) to FK
5C:  FX (L0 * M0) + FK to FJ
6A:  sum(MIN) to FW
6C:  FJ to FD; FW to FF
7B:  sum(MSUM) to FW
7C:  FF to L1; FW + FD to sum(MSUM)
```

Signum

```
0B:  if MIN is in modifier-this-pass quadrant, sum(MIN)
       to FG
0C:  sum(MRM) to FXA; M1 to FXB
1A:  if MIN is not in modifier-this-pass quadrant, sum(MIN)
       to FG
1C:  FG to FXA; M0 to FXB
4C:  FX (B * M1) to FK
5C:  FX (A * M0) to FJ
6C:  0 to FF; all-ones to FD if FJ .LT. FK
7B:  sum(MSUM) to FW
7C:  if FJ .GE. FK, FF to FR; if FJ .LT. FK, FD to FR;
       FW + FR + (1 if FJ .GT. FK, else 0) to sum(MSUM)
```

Threshold

```
0H:  L0 to FE
0B:  if MIN is in modifier-this-pass quadrant, sum(MIN) to
       FG
0C:  FE to FN; sum(MRM) to FXA; M1 to FXB
1A:  if MIN is not in modifier-this-pass quadrant, sum(MIN)
       to FG
1C:  FG to FXA; M0 to FXB
4C:  FX (B * M1) to FK
5C:  FX (A * M0) + FN (L0) to FJ
6C:  if FJ .GE. 0, FK to FF; if FJ .LT. 0, 0 to FF
7B:  sum(MSUM) to FW
7C:  FW + FF to sum(MSUM)
```

Inactive

```
6C:  0 to FF
7B:  sum(MSUM) to FW
7C:  FW + FF to sum(MSUM)
```

## Modifier Control

The modifier mode FMD0-4 is stored in 256x1 RAMs (MISCB, #1180), all addressed by CPABn. These mode bits are clocked through a pipeline FMDnBm (a '175 and a '174 on #1180). These are decoded by PROMs (on #1180 and #1181), addressed by FMD0Bn for ticks 0 and 1 and by FMD6Bn for ticks 6 and 7. PROM outputs and their meanings are as follows:

| | |
|---|---|
| FXM0P: | multiply in fraction mode |
| F1M1V: | 2 poles or 2 zeros, M1 variable |
| F1M0V: | 2 poles or 2 zeros, M0 variable |
| FGCA: | clock FG on phase B of even tick or phase A of odd tick |
| FGCB: | clock FG on phase C of even tick |
| FLA7P: | off if addressing L0 in first half of tick 1 |
| FLWCA: | write FL in second half of tick 0 |
| FLWHA: | write FL in first half of tick 1 |
| FEECC: | clock FEE on phase C of tick 0 |
| FEECH: | clock FEE on phase H of tick 1 |
| FPRSA: | select delay memory to FXA on tick 0 |
| FPGSA: | select FG to FXA on tick 1 |
| FPSGE0: | select FG or sum memory to FXA on tick 0 |
| FPSGE1: | select FG or sum memory to FXA on tick 1 |
| FQS0A: | select FV or FV shifted (A.M.) to FXB on tick 1 |
| FQS1A: | select 0 or FV shifted to FXB on tick 1 |
| FRKS: | select FK to FR on tick 6 |
| FYJS: | select FJ to FD on tick 6 |
| FYEA: | clear FF on tick 6 |
| FRFSA: | select FF to FR on tick 7 if FK .GT. FJ |
| FRFSB: | select FF to FR on tick 7 if FK .LE. FJ |
| FRFSC: | select FF to FR on tick 7 |
| FZS0: | select FC or FW to FF on tick 6 |
| FZS1: | select FW or FK to FF on tick 6 |
| FMDCP: | signum mode |
| FFRA: | clear FF on tick 6 if FJ is negative |
| FZCP: | zero-crossing pulser mode |
| FLWCB: | write FL on phase C of tick 7 |
| FLWCC: | write FL on phase C of tick 7 if FK .NE. 0 |

The FX multiplier scaling bits FXM1P and FXM2P come from RAMs on #1181. Two pairs of RAMs are 3-stated together, being enabled on alternate passes for the two successive multiplies of one modifier. Both enables are asserted when writing data into the RAMs. The bits are pipelined on #1182 to correspond to the data pipeline on the FILTB cards as previously described.

The PROM outputs, and in some cases mode bits taken directly, are combined with clock pulses as needed (see #1180, #1181) to implement the processing described in "Modifier Data Paths".

## Multipliers

There are four multipliers in the Synthesizer: GX, GY, GZ, and FX. Each is implemented in the form of a Wallace tree of four-bit slices. Because of the time required by a large tree, in the larger multipliers a pipeline is employed: partial products are formed on one tick and added together on the next tick.

Partial products are formed with 8875A and 8875B ICs and added by '283 adders. In a few cases carries are added together by 'H183s. Various portions of the four trees are allocated among the ten MULT cards, with a few remaining portions on the wire-wrap panels.

GX: this multiplier yields the low-order 13 bits of the product. The low-order 12 bits are generated by the Wallace tree, and the high-order bit by XORing the proper bits of the operands and carries out of the tree. As noted above, GX is modified to perform a function slightly different from simple multiplication.

GY, GZ: these are straightforward unsigned multipliers. The high-order part of the product is taken, but the low-order part of the tree is present to compute the proper carries into the high bits.

FX: this is the largest multiplier. It multiplies two 20-bit two's-complement numbers for a 39-bit two's-complement product. The signed result is generated in the Wallace-tree structure with the aid of three special types of PROM. One type takes in two four-bit numbers and produces the high-order four bits of their product, assuming that one of the operands is signed, in two's-complement form; the second type is similar but assumes both operands are signed. These are used in place of 8875As when the high-order four bits of either multiplier operand are involved. Corresponding PROMs for the low product bits are normal 8875Bs since the low bits are the same whether the multiply is signed or unsigned. The third special PROM type is used for sign extension, being added into a four-bit slice of the product, with its inputs coming from the high-order output bits of all signed-multiply PROMs in less significant slices. The 20-bit result FX0-19 can be selected from eight different positions in the 39-bit product. This is determined by mode bits FXM0-2, which control three successive stages of 'LS298s: FXG and FXH; FXJ; FX. The FXG and FXH partial products are added by '283s named FXI. Successive ticks in FX perform the following: 0 -- operands are clocked into FXA and FXB; 1 -- partial products are clocked into FXG and FXH, selecting between integer and fraction multiplication; 2 -- FXI is clocked into FXJ, selecting zero or two units of shift; 3 -- FXJ is clocked into FX, selecting zero or one unit of shift.

## Sum Memory

Sum memory is composed of 80 16x4 RAMs, organized in four quadrants named S0, S1, S2, and S3. Each quadrant is 64 words by 20 bits. Generator outputs are added in S0 and S1; modifier outputs in S2 and S3. On one pass, S0 will be "this pass" and S1 "last pass"; on the next pass, the functions will be exchanged. S2 and S3 alternate similarly. During a single tick, a quadrant may have as many as three separate read accesses or one read-pause-write access. These are interleaved as shown in Fig. 5. Two classes of modifier modes are distinguished: "mod-mix" modes which use the "A" operand early in their processing, and "pole-0" modes which use it later.

Sum memory activity is based on the clock phases PHA, PHB, and PHC. While a reference is occurring in each quadrant during a 65-nsec clock phase, the six-bit address in each quadrant is being generated for the next phase. On each phase, the addresses are clocked into 'S175s and 'S174s (center of #1184) producing S0A0-5, S1A1-5, S2A1-5, and S3A1-5. The inputs, SnAmI, are created on the four MISCA cards (left of sheet 2, #1184) by 'S153 multiplexers. The multiplexer selects, SnAS0-1, are based only on clock phases, OP, and OT. They are generated by gating on #1184 and by 'S51s on two SUM cards. Data inputs to the sum address multiplexers are as follows: FRM2-7, RAMs (on a MISCA) addressed by CP0Bn, holding modifier "B" addresses; GFM1-6, 93H72s on #1184, clocked every phase C from GFMI1-6, RAMs (on another MISCA) addressed by CPABn; FIN2-7, '175s on #1185 clocked every phase C from RAMs addressed on alternate ticks by CPABn and CP4Bn; and SUMA0-5, which is GSUM0-5 for quadrants 0 and 1 and FSUM1-6 for quadrants 2 and 3. GSUMn and FSUMn come from 'S161 counters on #1185; during processing ticks these counters are parallel-loaded on each clock phase from GSUMIn and FSUMIn, RAMs on MISCA cards addressed by CP6Bn through 'H04 inverters. Of the six address bits for a quadrant, the low-order four directly address the sum memory RAMs in the quadrant and the high-order two bits are decoded by 'S51s on various SUM cards to form the enables SnEm, where n is the quadrant and m denotes one of four banks of RAMs which are 3-stated together.

The high-order bits FRM0-1, GFM0, and FIN0-1 come from the RAMs as do the low-order bits, but instead of addressing sum memory they are used to control multiplexers which route sum memory outputs to the generators and modifiers. There are three sets of multiplexers, all on the SUM cards: SA0-19 (not labelled on the drawing), the generator fm input, formed by pairs of 'S257s; SB0-19 (not labelled), the generator sum term, formed by an 'S257 from quadrants 0 and 1 and an 'S258 from IRB0-29 in the generic interface for DMA read data; and SC0-19, the modifier input, two 'S258s from sum memory.

The signals that control these multiplexers are generated on #1184. SA23E, true when fm is coming from the modifier side of sum memory, is simply a buffered version of GFM0; SA1S to select quadrant 1 (as opposed to 0) and SA3S to select quadrant 3 (as opposed to 2) are just copies of OP. SBSE enables SB0-19 from sum memory, and SBIE enables it from the interface; these are opposite sides of a flip-flop ('S175 in 3E8) clocked from CHRP which indicates that the upcoming generator is in read-data mode. SB1S, selecting quadrant 1 rather than 0, is -OP buffered. The multiplexer controls for SC0-19 involve OP and two signals, CSS0-1, coming from #1172, which sequence through the various phases of even and odd ticks. The CSSn address an 'S153 multiplexer to select SCSP0-1 from the high-order bits of FIN, FRM, or FSUM. The SCSPn are combined in gating and clocked into flip-flops on every clock phase to form SC01E (enable from generator side), SC23E (enable from modifier side), and SC3S (quadrant 3 as opposed to 2). SC1S, selecting quadrant 1 as opposed to 0, is OP buffered.

Resetting sum memory is governed by flip-flop SR on #1184. Its D input is SRI (#1173), arranged so that SR will set after the first interface tick of a pass and will clear by processing tick 6. SR is ANDed with OPD (OP delayed -- see Fig.2) and its complement to give SR0 (reset even quadrants) and SR1 (reset odd quadrants). These are ANDed with CU (clock on phases A, B, and C) by 'S51s on SUM cards, to assert chip enables and write pulses for all RAMs in the appropriate quadrants. While SR is asserted, the GSUMn and FSUMn 'S161 counters on #1185 are conditioned to count on every clock phase, disabling the parallel entry. When they have counted through 16 states (less than 6 ticks), sum memory has been reset.

Normal writing into sum memory is controlled by the write grant signals SnWG (#1184). For the modifier quadrants these are the AND of OT, CTR7, and OP or -OP. The generator write grants do not involve OT but include a generator mode bit which governs adding to sum memory.

## Delay Memory

The DMEM cards (#1121) use 4096-bit dynamic MOS RAMs. Each card has a 21x4 array of RAMs comprising 16K 20-bit words plus parity. Addresses, write enable, column strobe, and chip enable are buffered by '128s and series-terminated with 33-ohm resistors. There are six address lines, time-multiplexed to give row address and column address in sequence. Input data, RT0-19 and RTP (parity), are buffered in 'LS174s clocked on the high-going transition of RMCOL. Output data goes through 'LS365 buffers, enabled by RMDSn, which gate data from the proper DMEM card onto the 3-state bus RR0-20. Parity is generated (RTP) and checked (RR20) with 9348s on INTF cards (#1122). The row strobes RS0-3 determine which row of chips on the DMEM card actually perform a given cycle. RSn on board m is the AND of the board select RMBSm and the row select RMRSn.

Control signals for the DMEM cards are generated on #1188. There are three types of cycles: normal (i.e. delay unit), refresh, and PDP-10 access, associated with the control signals RMNY, RMRY, and RMTY respectively. Each such signal is true during the four ticks of a cycle of the proper type. An 'LS195A shift register, clocked on phase C, counts the four ticks of a cycle: RMC0 is true in the first tick; RMC0 and RMC1 in the second; RMC0-2 for the third; RMC0-3 all true for the fourth. Another 'LS195A, clocked on the leading edge of phase C, provides the timing signals which after gating are used on the DMEM cards: RMRAS (row strobe), RMCAS (column strobe), and RMWPP (write pulse). An 'H74 clocks in RMRAS on phase A to create the RMCOL signal. Priority arbitration for the next cycle is done by gating at the input of the 'LS175 which generates RMTY, RMNY, and RMRY. A PDP-10 access has the highest priority and its request line goes right to RMTY. If there is no PDP-10 request, a request for a normal cycle (RQP) on a processing tick (CTRA) turns on RMNY. Failing both those conditions, a refresh request (RMRQ) sets RMRY.

The cycle-type flags are only clocked on ticks in which RMI is true, indicating a cycle is about to end or none is in progress. This signal also resets the 'LS195As. The normal cycle request RQP is an 'LS109 conditioned by clock enables to permit only one normal cycle per gated clock tick. (Most of the DMEM control logic runs on ungated clocks, since refresh and PDP-10 cycles must be permitted and normal cycles completed once begun, even if the clock is stopped.) RQP is held off by RQO, which is overflow from the counter RQ0-4 ('LS161s). The RQn counter is reset at the beginning of each pass and counts the 32 delay units. Its trickle enable, RQET, is asserted during the third tick of a normal cycle. The refresh request flip-flop RMRQ ('LS109) is set by RMRT, trickle carry out of a 7-bit counter ('LS161s), which counts out and requests a refresh cycle approximately every 24 microseconds. The RMRQ flip-flop provides one level of buffering for RMRT so that a second refresh request can be timed out while one is pending. A refresh address counter RMR0-5 ('LS161s) is advanced at the end of each refresh cycle. It runs through all 64 states to ensure that all row and column addresses in the dynamic RAMs are refreshed in turn. The 16-bit delay memory address RV0-15 is treated as follows: RV0-3 are latched in an 'LS157 as RMA0-3. Then RMA0-1 are decoded by an 'LS139 to form the board selects RMDSn and RMBSn (through an 'LS158 to make them row strobe pulses, all on for refresh cycles). RMA2-3 are decoded by another 'LS139 section to form the row selects RMRSn. A set of 'LS153 multiplexers form the six address bits RMA4-9 sent to the RAMS directly; these select either RV4-9 for the row address, RV10-15 for the column address, or RMR0-5 for both in refresh cycles.

## Delay Memory Data

The DMD cards (#1120) contain the delay unit data paths other than the delay memory itself. The general organization is shown in Fig. 6. There are seven 32-word memories (one word per delay unit): RA0-19, RB0-19, RC0-19, RD0-19, RX0-19, RY0-19, RZ0-19. The RA, RB, RC, and RD memories are used for interfacing to the modifiers. On even passes, modifiers write into RB and read from RD while delay memory is being read into RC and written from RA. On odd passes RA and RB exchange functions as do RC and RD. The memories are addressed through '157 multiplexers (#1187) selected by OP. The address of the memory sending data to the modifiers is FRM3-7 (discussed above); for the memory receiving data from the modifiers, RG0-4, which is FRM3-7 delayed by 'LS670s; for the memory sending data to delay memory, RQ0-4, the delay unit counter; for the memory reading data from delay memory, RK0-4 (an 'LS174 on #1188), a delayed version of RQ0-4. RX, RY, and RZ are addressed by RS0-4 (RS0-3 are buffered by 'H04s on the DMD cards) which is generated by multiplexers (a 9309 and an 'LS158) on #1187. The multiplexer inputs are: RQ0-4, the delay unit counter; and IC7-11, command bits from the generic interface, which is selected during update ticks. RX0-15 has the base address in delay memory; RY0-15 (in delay line mode) has the index into delay memory; RZ0-15 has the limit in delay line mode, RZ12-15 has the scale factor in table look-up mode; RX16-19 has the mode; RX16-19 and RY16-19 are unused.

For PDP-10 cycles, data to be written is selected at the RT multiplexers by RTS, and the address is selected at the RV multiplexers by RVS; both selects are simply copies of RMTY (#1187). At the end of the cycle, RLC clocks the RR bus into RL0-19, which can be read by the PDP-10 interface.

For normal cycles in delay-line mode, the information to be written is taken from RA or RB by the RAB0-19 multiplexers ('LS157s on DMD) and goes through RT. The address is generated by '283 adders which add the outputs of the RX and RY RAMs. At the same time, RY is incremented by 1 ('283s) and the result clocked into RW0-15 ('LS175s). The unincremented RY is also being compared to RZ by '85s. On phase C of the third tick, RY is written (by RYW) from the RU0-15 multiplexers ('LS158s, outputs not labelled): either the incremented value (RW) or zero will be written, depending on RUE which will be asserted (to enable RW) unless RY = RZ. The RUE gating is on #1187. On phase C of the fourth tick, RC or RD is written (RCW or RDW, gated from RCDW on #1188) from RR0-19.

For normal cycles in table look-up mode, the base address RX is taken as before, but the RY RAMs are disabled (RYEA and RYEB held false) and the 'LS253 RY multiplexers are enabled (RYEC true) (gating from the mode on #1187). The RO and RY multiplexers are selected by the scale factor RZ12-15 to apply 0 to 15 units of shift to RAB. A low-order bit, RY16, is generated on #1187; if it is 1 and the rounding mode is specified, RVIC16 is asserted to inject a carry into the '283s adding RX and RY. Writing of RC or RD is as above.

The RI0-19 lines, from 'LS157 multiplexers on the FILTB cards, carry both data from the modifiers to be written in RA or RB and data from the interface to be written in RX, RY, or RZ. The selection is governed by RIS (#1181), which is a copy of ITR. RI0-19 appear directly on the data inputs to RX and RZ and, through the RU multiplexers, to RY. RI is the input to the RE0-19 register, 'LS175s on DMD clocked on phase C, which in turn is written into RA or RB (by RAW or RBW, gated on #1187 with FPRD, to indicate that the current modifier is in delay mode).

## Generic Interface

The principal feature of the generic interface, shown in block diagram form in Fig.7, is the 32x32 FIFO, designated IF0-31, comprised of 64-bit RAMs on the INTF cards. This holds up to 28 commands and 4 read-data items. Commands come out of IF into the '175 registers I0-19 and IC0-11; read data go to the 'LS175s forming IRB0-19 (on the SUM cards). Input to the FIFO is from the 16-bit register IM0-15 ('LS175s on INTF). The FIFO can be addressed from four different counters: IFIC0-5 for command input; IFOC0-5 for command output; IFIQ0-2 for data input; IFOQ0-2 for data output. Each counter is advanced at the conclusion of the relevant type of cycle. The high-order bits do not actually address the FIFO but are used to distinguish full and empty conditions: if all input and output counter bits agree, the buffer is empty; if all but the high-order bits agree, the buffer is full. The data counters go from 0 to 3; the command counters from 4 to 31 decimal. All are formed of 'LS161s on #1190. On a given tick, the FIFO can do either a 32-bit output cycle or a 16-bit input cycle, according to the following priority scheme:

A -- On processing ticks:

- A1 -- If possible, FIFO to IRB
- A2 -- Else if possible, IM to FIFO (left half, then right half)

B -- On update ticks:

- B1 -- If possible, FIFO to I and IC
- B2 -- Else if possible, IM to FIFO (left half, then right half)

Here "if possible" means if the source has data and the destination has room to accept data of that type.

Flip-flop IMBF ('109 on #1190) is set when the PDP-10 interface puts information into IM, and cleared on a right-half IM-to-FIFO cycle. Similarly, IRBF is set by a FIFO-to-IRB cycle and cleared when the generator calculator takes the data from IRB.

The priority scheme is implemented in a straightforward though lengthy manner with a comparator, gates, and one PROM. The comparator develops IFCNF, "IF commands not full". The PROM generates IFQNF ("IF data not full") and IFQAV ("IF data space available"); the difference is that in packed data mode (determined by TP7), there must be two data slots free in the FIFO before initiating a data read. In such a case IFQAV is more cautious than IFQNF. The combinational logic develops IIL, which calls for a B1 cycle, and IRBCA, which calls for A1. Final outputs are IFAS0-1, which are the selects on the FIFO address multiplexers IFA0-4 (a '157 and a 9309 on #1190).

| IFAS0 | IFAS1 | Address | Type of cycle |
|---|---|---|---|
| 0 | 0 | IFIQ1-2 | data to FIFO |
| 0 | 1 | IFOQ1-2 | FIFO to IRB |
| 1 | 0 | IFIC1-5 | command to FIFO |
| 1 | 1 | IFOC1-5 | FIFO to I, IC |

The type of information in IM is encoded by IMT0-2, which are set up by the PDP-10 interface.

| IMT0-2 | Data in IM |
|---|---|
| 000 | packed data right half |
| 001 | packed data left half |
| 010 | unpacked data right half |
| 011 | unpacked data left half |
| 100 | (invalid) |
| 101 | (invalid) |
| 110 | command right half |
| 111 | command left half |

For writing packed data into the FIFO, an 'LS157 on DMD introduces 0 in the low-order bits if IMT1 is false.

Command decoding is shown on #1191. IILD (a '109 on #1190), true when a command is present in I and IC, and -ITR are used to enable 'LS138 decoders whose outputs (of the form IIMn) identify the various commands. Various commands have effect within the generic interface: IIM02 and IIM03 are gated (#1190) to form ITAC and ITBC, which clock TTL buffers A and B, respectively -- 'LS157s on INTF -- whose outputs are followed by 7437 buffer gates. The register IX0-15 ('298s on INTF) is clocked by IXC (from an 'H51 on #1190) either due to IIM01 or on the tick after an IIM1, IIM2, or IIM6 with IC3 true; in the latter case, zeros are loaded into IX, corresponding to the "clear DX" function of certain commands. IIM21 loads IP0-19 ('LS163 counters, IP0-15 on INTF cards and IP16-19 shown in #1190), the pass counter. The counter is enabled by EPAS.

The commands to clear all wait bits and to clear all pause bits use flip-flops IRW and IRP ('H74s on #1190). Each is set by PHBM (phase B of an update tick when performing a "miscellaneous" command, generated on #1191) if the appropriate command bit is set in IC. They are clocked off at the end of the next pass's real (processing) ticks.

For the case of a linger command, I0-19 is always being compared to IP0-19 ('283s performing subtraction followed by gates to indicate a zero result). ISOK (meaning linger satisfied) is generated (on #1190) if the pass count is at least equal to the command data and does not exceed it by 4,096 or more; ISU (underrun) is true if ISOK is true and the numbers are not exactly equal. The IIM23 version of linger clears the pass counter first (signal IPR on #1190); repeated clearing is inhibited by the IPRD flip-flop which remains set until the linger is finished. This condition contributes to the term IILF (#1190) which indicates that a new command can be loaded into I and IC. The intermediate term IIMW (#1190) is true when a linger command is present and the pass counter is valid (not being reset).

The interface address, IA0-7, is the generator or modifier number of the next command to be performed. If it is a generator number, it contains all 8 bits; a modifier number occupies the left 7 bits with the rightmost bit distinguishing L0 from L1 or M0 from M1. The address must be available the tick before the command is executed, so that it can be loaded into the clock pipeline. Therefore the IA multiplexers ('LS257s and 'LS258s on #1190) have four possible sources: two (generator or modifier number) from IF, to load the address into the CPnBm at the same time the command is loaded into I and IC; and two from IC, similarly, for a command previously loaded into I and IC but not yet performed. The enables and selects IAEA, IAEB, IASA, and IASB perform this function.

Data being written to computer memory is put in IWB0-19 ('LS298s on SUM cards) by the generator calculator, which at that time sets IWBF ('109 on #1190). The PDP-10 interface clears IWBF at the end of a DMA write cycle.

## PDP-10 Interface

Data paths for the PDP-10 interface are on the nine TENI cards. The low-order six cards, termed "full", have both data and memory address logic; the high-order three cards, termed "partial", have data but not address logic. A block diagram is given in Fig. 8.

The DEC-level I/O and memory busses are received and converted to TTL by 75110 differential line receivers, biased to VREF (about -1.5 volts). The memory data, being in pulse form, is "caught" by 9314 latches to form TD0-35. Parity of each four-bit slice is generated on the TENI card by 'LS86s; the results are then combined to form TDEP by a 9348 on #1193.

The busses are driven by 75110s followed by 2N4258 transistors. In keeping with PDP-10 practice, the I/O bus lines are driven to ground, and the memory lines toward -5 volts with a 100-ohm parallel termination to ground. Data to memory bus bits 4-23 comes direct from IWB0-19; zeros are written for the other bits. The parity bit, IWBP, is formed by 9348s on #1184. The I/O bus bits 16-35 are driven from the lines TTT16-35, coming from 'LS153 multiplexers that select one of four sources: D0-19, the diagnostic bus; RL0-19, the delay memory output register; various bits for CONI-A; and the TCC register (slightly scrambled), for CONI-B. I/O bus bits 0-15 are not driven. The received I/O bus appears on the output of 'LS157 multiplexers as TR0-35 (the other multiplexer input is currently unused); this is clocked, in the case of DATAO, into the 'LS175s TH0-35. Data passes from the PDP-10 interface to the generic interface (IM register) through 'LS257s on the INTF card. These select between left half (bits 4-19) and right half (bits 20-35) of either TH (DATAO data) or TD (data read from PDP-10 memory). TH0-35 also go to the delay units for PDP-10 references to delay memory.

On the full TENI cards is a 16x24 memory called TCI0-23. It is addressed by TAA0-3 (a '157 on #1195). Six words of this memory are actually used, as follows:

| Address | Contents |
|---|---|
| 1 | write data CA |
| 2 | read data CA |
| 3 | command CA |
| 5 | write data WC |
| 6 | read data WC |
| 7 | command WC |

TCI0-23 is clocked into two different registers, both comprised of 'LS175s on the TENI cards: TA14-35, which is the data actually put on the address lines of the PDP-10 memory bus; and an unlabelled register whose output is incremented by 1 in a set of '283s to form TC0-23. Associated with each WC is a flip-flop ('LS109) on #1195: TWE for write data, TRE for read data, TCE for commands. These are the "exhausted" flags for the three types of DMA; they are directly set by the master reset function.

When a DATAO-B is performed, TH0-3 (the high-order four bits of the DATAO data) are selected onto TAA0-3, and TH12-35 are selected for the data input to the TCI RAMs (selection is by 'LS158s on TENI). The select signal, TAWW, comes from an 'H30 on #1195 and indicates not only that a DATAO is occurring but also that no memory cycle is in progress. It is used to turn off the proper "exhausted" flag, as selected by TAAS1-3, a phase C clock pulse decoded from TAA0-3 by an 'LS138.

TAW, the write pulse to the TCI RAMs, occurs on phase C ('H00 on #1195).

Addressing of a memory cycle proceeds through four successive ticks, indicated by one of the following in succession being true: TMG, TMGD, TMGDD, TMG3D. On the first tick, the appropriate CA word is addressed in TCI; phase C clocks it into TA to drive the PDP-10 memory address lines, and also into the other register to be incremented. On phase C of the second tick, the incremented CA is written back into TCI. On the third tick, the relevant WC is clocked into the unnamed register; on the fourth tick the incremented WC is written back into TCI and, if a carry comes out of the incrementer, the proper "exhausted" flag is set.

Control of the actual memory cycle is shown on #1193. Three 'LS10 gates create cycle requests TRY, TWY, and TCY (read data, write data, and commands) according to the "exhausted" flags and buffer-register or FIFO full or available flags. When a memory cycle is not in progress any request forms TMQ, which is clocked into an 'LS174 (on #1195) and comes out as TMY. This is turn becomes TMG ("memory go") if no cycle or DATAO is in progress. TMG is clocked into TMB ("memory busy") to indicate a cycle in progress. At the same time, TWQ is clocked into TMW if a write cycle is requested. TMB asserts TMRQ, the memory port multiplexer request. The acknowledgement TACKN asserts TAE, enabling the driving of the memory address lines, and also, after a 50-nsec delay (SNG82), putting up TREQCYC to indicate a memory cycle request. The memory responds with TMAA (address acknowledge) which resets the TD latches if in a read cycle, and is latched as TMA which turns off TMRQ. TREQCYC was turned off by TMAA and is held off by the absence of TMRQ. On a write cycle, the fall of TMAA initiates an 85-nsec delay (SNG82) during which TWE is on to enable driving the data bits of the memory bus. When the delay expires, TMDS sets TMD which is resynchronized as TMDD. TMDD causes TMF which clears TMB to end the cycle.

A read cycle proceeds as above through the address acknowledge pulse. Then TDR clears the pulse-catcher TD0-35 and TMA then enables TD for input (TDE). The read restart pulse TRDRS eventually clocks on TMD, turning off TDE. If there is no parity error, -TDEP and TMDD then finish the cycle. If a parity error occurs, the memory control stays in a busy state with TMPE on until the problem is acknowledged by a CONO-B to reset it. Such a CONO sets TMC ("memory continue") which permits the cycle to finish, asserting TMER to reset the error flip-flop. To detect nonexistent memory, a counter ('LS163s) counts about 50 microsec after the start of a cycle and then asserts TMXO, which, if the memory is still busy, sets TMXE. This state is also gotten past by a CONO-B which sets TMC.

The I/O bus interface begins with the device address comparator, a 9324 on #1194, which asserts TS if the high-order five IOS bits (TRS3-7) agree with five bits set by switches TSS3-7. TS and either DATAI or CONI create TTE which enables the bus drivers on the TENI cards. TS is ANDed with DATAO-CLR to give the clock TDOC. The low-order IOS bits, TRS8-9, are used to decode TSA (device code A selected) and TSB (for device code B), and are clocked by TDOC into a '175 to emerge as TMDOR and -TMDOA. CONO-A, CONI-B and DATAOs are synchronized in an 'LS175. In the case of a DATAO, the TMDO flip-flop is set and remains set until the function of the particular DATAO has been performed. Its K term comes from an 'LS20 whose four inputs correspond to completion of the four DATAOs:

| | |
|---|---|
| IWBG | DATAO-C |
| RTD | DATAO-D |
| TMDOW | DATAO-A |
| TAWW | DATAO-B |

CONO-A, indicated by the unsynchronized clock TCOCA, sets TIA0-2 and TIB0-2 ('LS174s on #1195) according to the CONO data; these are the PIA numbers. It also clocks the diagnostic address from TR25-31 into IDA0-5 and IDS (a '174 and a '175 on #1194). The synchronized version TCOCADA clocks IRH ('LS109) which is on in "all ticks update" mode.

CONO-B uses bits TR32-35 to address two 9334 addressable latches, writing bit TR31 into the addressed cell. The latched outputs, TP0-15, are the BB.AAA conditions of the programming specification.

The master reset signals MR and TR have three causes: IOB RESET, called TMR; power-on reset, POR; and the reset bit of CONO-A.

The TCC register is read by CONI-B. This is formed mostly of 'LS175s on INTF cards; but bit TCC5, the lost cause bit, is an 'LS109 on #1195. The register is clocked from TJ0-4 and CTK1-8. The CTK lines are the current time state; the TJ lines, generated on #1191, represent in order the I1-5 bits of CONI-B. TCC is clocked by TCCC each phase C until a bit appears in it which is masked on by an 01AAA CONO-B bit (TP9-13). Such a condition asserts TCCF (#1195), which suppresses TCCC. If on a subsequent tick a TJn masked on by TP9-13 occurs, ITJ (#1191) will be true and TCC5 will be turned on. TCC5 is cleared by TCIBR which comes from CONI-B (#1194).

The interrupt request lines TIOBPI1-7 come from a 9334 addressable latch on #1195. Twice per tick a bit in the latch is strobed in; on the first half of the tick from TIA (interrupt request, channel A), and on the second half from TIB; address is TIA0-2 or TIB0-2 respectively. The interrupt requests are formed from the various interrupt conditions and the mask bits, TP6 and TP8-14. All requests are reset by the master reset function, and by any DATAO, CONO-A or CONO-B.

Analog Outputs

Each ALOG card contains two independent analog output channels. All channels share the same clock (phase A) and data inputs (AP0-13), differing only in their SEL signals. These are the enables AE0-15 decoded from GO16-19 by 9301s on #1175, including the GD term indicating a generator in DAC mode.

A 14-bit register, shown on the ALOG schematic (#1110) as D0-13 ('LS174s), holds the current DAC input. The SAMP flip-flop ('LS109) is true when the sampling switch is in the sample state and false when it is in the hold state. A counter ('LS109 and 'LS163) times the duration of the hold state.

The DAC has a built-in op amp which converts the current switch outputs to a low-impedance voltage source. R35, combined with internal feedback and offset resistors, adjusts the DAC output range to + or - 1.5V. This is applied to a sample-and-hold circuit whose output, SIG, drives the filter chain or final output amplifier. The sampled DAC value is stored in C14, which is buffered by U10 to drive SIG. Q5 acts as a switch which, when closed, allows C14 to charge to the DAC voltage through R3 and R39.

About 260 nsec before the DAC register is updated, the SAMP signal goes high (false), cutting off Q1 and allowing Q2 to turn on. This turns on Q3 which applies -15V to the gate of Q5 through CR3 and CR4, thereby cutting off Q5. Approximately 6 microsec later, after the DAC output has settled to its new value, SAMP falls, allowing +15V to be applied to the cathode of CR4, permitting Q5 to turn on. Some of the switching voltage is transferred to C14 because of the capacitance of Q5. C13 couples a pulse of opposite polarity and approximately the same amplitude into C14 to minimize this effect.

An 'LS175 holds FE (filter enable) and FA0-1 (which determine filter frequency). FE drives a DG154 FET switch which selects filtered or unfiltered output; FA0-1 operate other switches which program the UAF31 active filters. The filter configuration is 6-pole Butterworth, with breakpoints as follows:

| FA0-1 | freq |
|---|---|
| 0 | 4.5 kHz |
| 1 | 9.0 kHz |
| 2 | 13.5 kHz |
| 3 | 18.0 kHz |

The FE and FA 'LS175 is loaded at the same time, with the same data for all channels, by AFC (#1175) which derives from IIM03.

## Diagnostic Readback

The high-order 6 bits of the diagnostic address, IDA0-5, are decoded by 'LS138s on #1197, to form diagnostic enables DE00-63 (numbered in octal). Each enable goes to a group of up to five 'LS257s (or 'LS258s) whose outputs form the 3-state diagnostic bus D0-19. The low-order address bit IDS is buffered to perform the selection on all the diagnostic multiplexers. Contents of the several addresses are tabulated below.

Left Column: Diagnostic Address (in octal)

Right Columns: 20-bit Diagnostic Word (bit numbers in decimal), X=undefined bit

0 TTT0-19

1 TC4-23

2 X X X X X X TIOBPI0-1 X X TIOBPI2-3 X X TIOBPI4-5 TTT12-13 TIOBPI6 X

3 X X X X X X X X X X X X X X X TC0-3

4 TD16-35

5 TH16-35

6 X X X X TD0-15

7 X X X X TH0-15

10 TA4-23

12 X X X X X X X X X X X X X X X X TA0-3

13 X X X X IWBP X X X TMRQ TMPXCLR X X TREQCYC TMW X X X X X X

14 IX0-15 X X X X

15 TCC0-4 TCC6-16 X X X X

16 II16-19 IC0-11 X X X X

17 IM0-15 X X X X

20 IP0-15 X X X X

21 II0-15 X X X X

22 ITB0-15 X X X X

23 ITA0-15 X X X X

24 GT0-11 X X X X X X X X

25 GL0-11 X X X X X X X X

26 GZA0-11 X X X X X X X X

27 GZB0-11 X X X X X X X X

```
30 GXC0-10 X X X X X X X X X
31 GYA0-3 GYC0-3 GYE0-3 X X X X X X X X
32 X X X X GY4-11 FRM4-7 GSUM2-5
33 X GX1-11 FRM0-3 X X GSUM0-1
34 GYB0-11 CP4B0 CP5B0 CP6B0 CP7B0 CP4B2 CP5B2 CP6B2 CP7B2
35 GYD0-11 CP4B1 CP5B1 CP6B1 CP7B1 CP4B3 CP5B3 CP6B3 CP7B3
36 GWD0-11 FSUM3-6 GFMI3-6
37 GU0-11 X FSUM0-2 X GFMI0-2
40 GXB0-11 CP4B4 CP5B4 CP6B4 CP7B4 CP4B6 CP5B6 CP6B6 CP7B6
41 GWB0-11 CP4B5 CP5B5 CP6B5 CP7B5 CP4B7 CP5B7 CP6B7 CP7B7
42 GVB2-12 X X X X X X X X X
43 GJ0-7 GVA0-3 X X X X X X X X
44 GAZ0-7 GC0-3 X X X X X X X X
46 GM0-3 GRMD0-3 FMD0-3 X X X X X X X X
47 X X FU0-9 X X X X X X X X
50 FE0-19
51 FEE0-19
52 FXA0-19
53 FXB0-19
54 FW0-19
55 FG0-19
56 FU10-29
57 FQI0-19
60 FV0-19
61 -GRMD0B0-3 -GRMD1B0-3 -GRMD3B0-3 -GRMD5B0-3 GRMD6B0-1 GRMD7B0-1
62 FN0-19
```

63 TP0-15 X X X X
64 FF0-19
65 FD0-19
66 FJ0-19
67 FK0-19
70 FXJ0-19
71 FX0-19
72 FXH0-19
73 FXH20-23 FXG6-7 FXG10-11 FXG14-15 FXG18-19 FXG21-23 FXH24 FXG3 X X X
74 AP0-19
76 S0B0-19
77 S1B0-19
100 S2B0-19
101 S3B0-19
102 GRA0-19
103 IWB0-19
104 GRB0-19
105 -IRB0-19
106 GWA0-19
107 GB0-19
110 GXA0-19
111 GK0-19
112 GVA4-23
113 GJ8-27
114 GAZ8-27
115 GC4-23

```
116 GP0-19
117 GO0-19
120 S0A0-5 S1A0-5 SCS0-1 SA23E SR OPD GFM0-2
121 S2A0-5 S3A0-5 X X X X GFM3-6
122 GXB12 GWB12 -CSUM0-5 IA0-3 X CPP CTR1F CTR4F CTRO CTRA ITR OT
123 FIN0-7 IA4-7 CTR0-7
124 CPAB0-7 X X CTK0-9
125 CTP0-9 CTT0-9
126 FXM1P FXM2P FMD4 FMD6B0 IRP IRW IMBF IWBF IP16-19 IFIQ1-2 IFIC0-5
127 FMD6B1-4 IIL IILD -IFAS1 -IFAS0 X X IFIQ0 IFOQ0-2 IFOC0-5
130 GZ0-17 GZC3 GZCN
131 GWBO GWCO GEC GUF GY0-3 GVB0-1 GVB13 GX0 GJ0D GRA0D FLIDFEA X X X X GYQ3
132 RMRY RMNY RMTY RMC0 RP0-1 RMC2 RG0-4 RPE RQO FPRD RQ0-4
133 RMRAS RMCAS RMWPP RMC1 RP2-3 RMC3 RK0-4 RMRT RMRQ RMR0-5
134 IDA0-3 GXS GXBS GUS0-1 X X X X GRMDA0-3 TIA0-2 RY20
135 IDA4-5 X IDS GX12 X X X X X GINVBE GTS GRMDA4-7 TIB0-2 X
136 IMT0-2 TMW TMB TMC TMD TMDD TZA X TCC5 X TMDO TMDOA TMDOR TDOCD TMGD TMGDD TMG3D TMXO
137 X X X X -TMY TAAI0-1 X TDOCDD TCOCAD TCOCADD X IRH TCE TRE TWE GTT GTA GCOD GCOG
140 RE0-19
141 RT0-19
142 R0-19
143 RV0-15 X X X X
144 RX0-19
145 -RW0-19
146 RY0-15 X X X X
147 RZ0-19
```

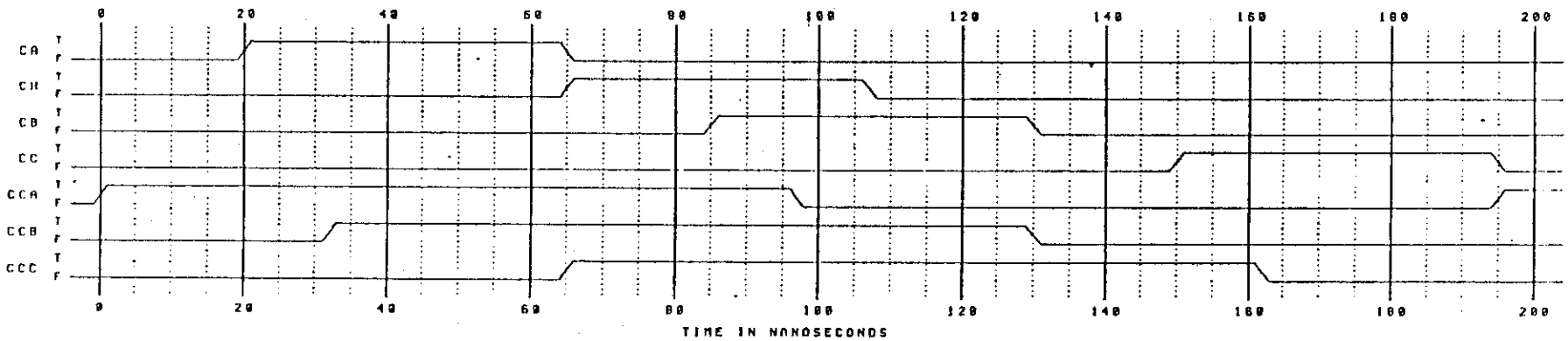



Figure 1. Clock Phases

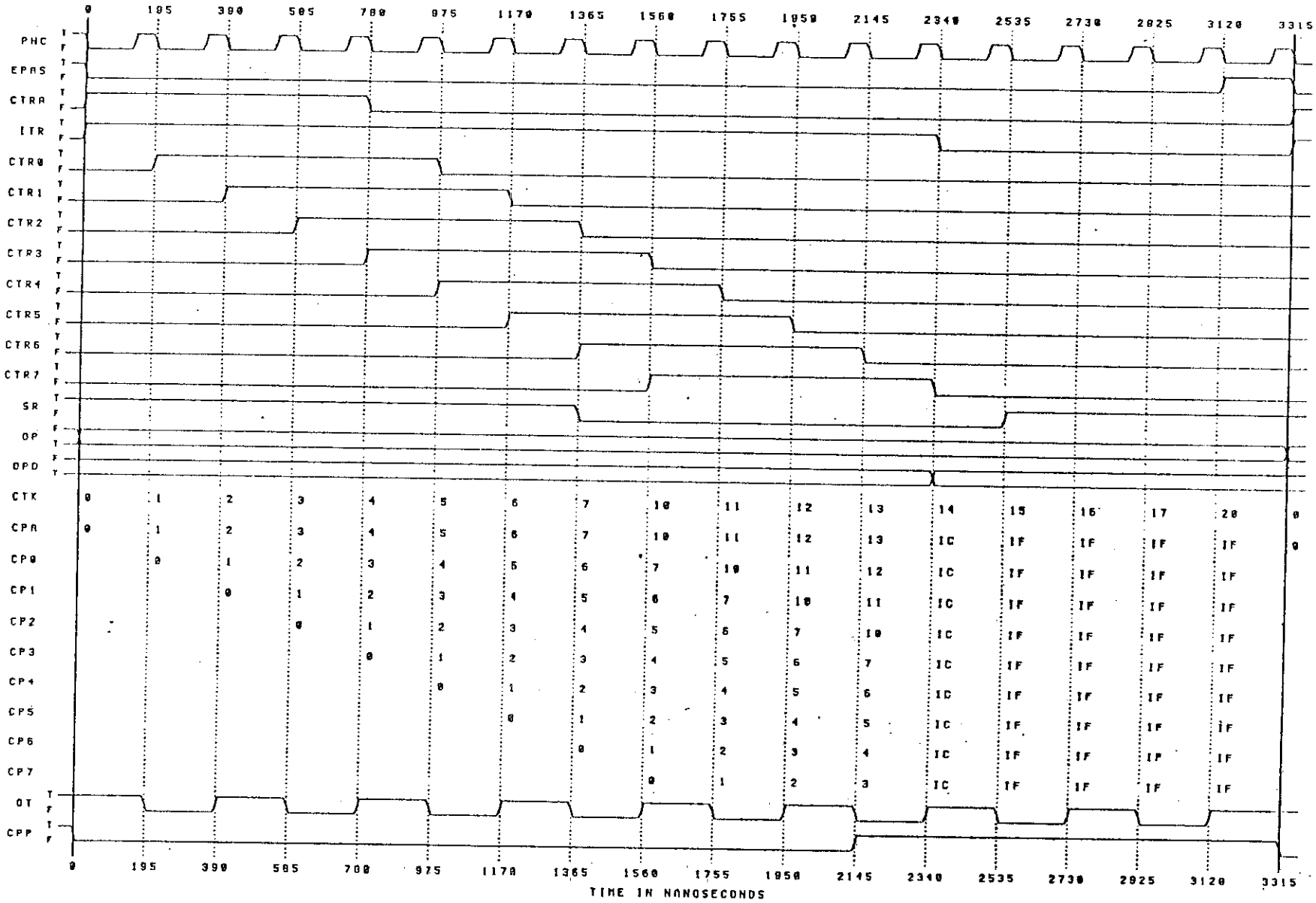



Figure 2. Clock Counting

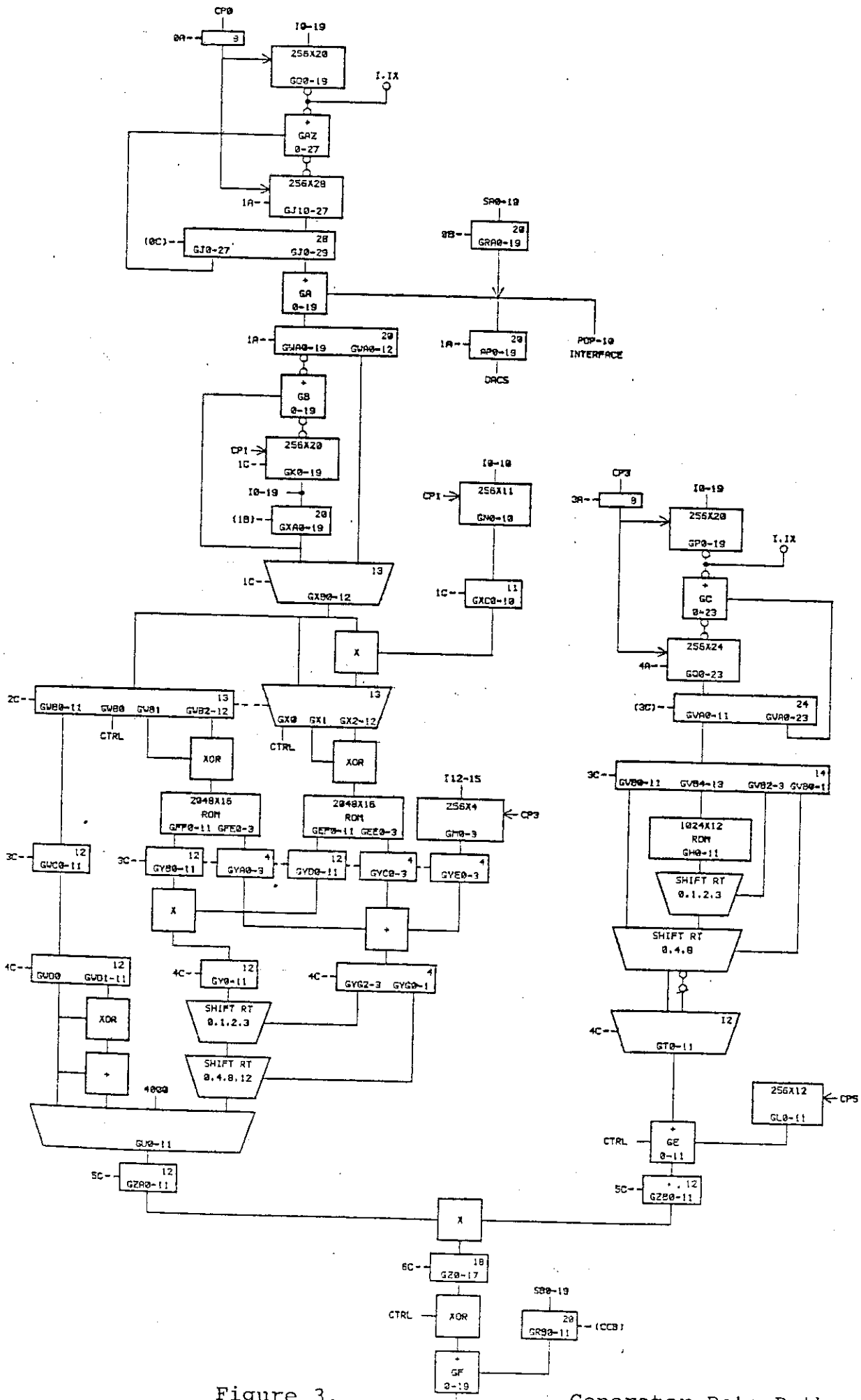

CP0
0A
9
I0-19
256X20
GD0-19
I.IX
GAZ
0-27
256X28
1A
GJ10-27
(0C)
GJ0-27
28
GJ0-29
SA0-19
0B
20
GRA0-19
GA
0-19
1A
GVA0-19
20
GVA0-12
1A
AP0-19
20
DACS
POP-10
INTERFACE
GB
0-19
CP1
1C
256X20
GX0-19
I0-19
(1B)
20
GXA0-19
I0-10
CP1
256X11
GN0-10
1C
GXB0-12
13
1C
GXC0-10
11
X
CP3
3A
9
I0-19
256X20
GP0-19
I.IX
GC
0-23
256X24
4A
GD0-23
(3C)
GVA0-11
24
GVA0-23
2C
GVB0-11
GVB0
GVB1
GVB2-12
13
CTRL
XOR
GX0
GX1
GX2-12
13
CTRL
XOR
I12-15
2048X16
ROM
GFF0-11 GFE0-3
2048X16
ROM
GEF0-11 GEE0-3
256X4
GH0-3
CP3
3C
GVC0-11
12
3C
GYB0-11
12
GYA0-3
4
GYD0-11
12
GYC0-3
4
GYE0-3
4
X
+
3C
GVB0-11
GVB4-13
GVB2-3
GVB0-1
14
1024X12
ROM
GH0-11
SHIFT RT
0.1.2.3
SHIFT RT
0.4.8
4C
GVD0
GVD1-11
12
4C
GY0-11
12
4C
GYG2-3
GYG0-1
4
XOR
SHIFT RT
0.1.2.3
+
SHIFT RT
0.4.8.12
4C
GT0-11
12
4000
256X12
GL0-11
CP5
GV0-11
CTRL
GE
0-11
5C
GZA0-11
12
5C
GZB0-11
12
X
6C
GZ0-17
18
S80-19
CTRL
XOR
GR80-11
20
(CCB)
GF
0-19

Figure 3.

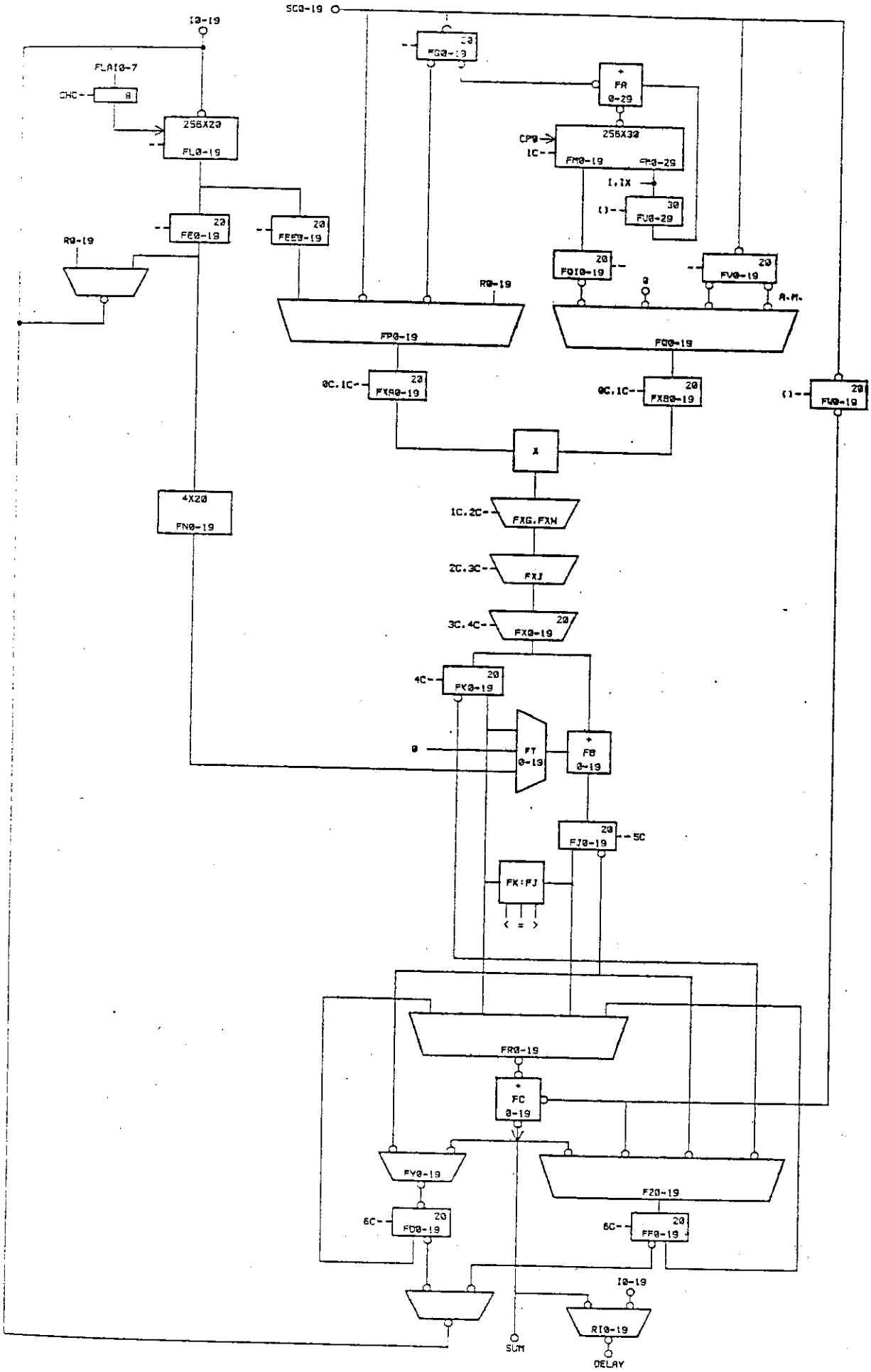



Figure 4. Modifier Data Paths

| | EVEN TICK | | | ODD TICK | | |
|---|---|---|---|---|---|---|
| | phase A | phase B | phase C | phase A | phase B | phase C |
| S0 | pole-0 read IN | generator read FM | modifier read RM | mod-mix read IN | generator read FM | |
| S1 | generator read - pause - write SUM | | | generator read - pause - write SUM | | |
| S2 | pole-0 read IN | generator read FM | modifier read RM | mod-mix read IN | generator read FM | |
| S3 | pole-0 read IN | mod-mix read IN | modifier read RM | | modifier read SUM | modifier write SUM |

EVEN PASS SHOWN (OP false)

EXCHANGE S0 WITH S1 AND S2 WITH S3

FOR ODD PASS (OP true)

Figure 5. Sum Memory References

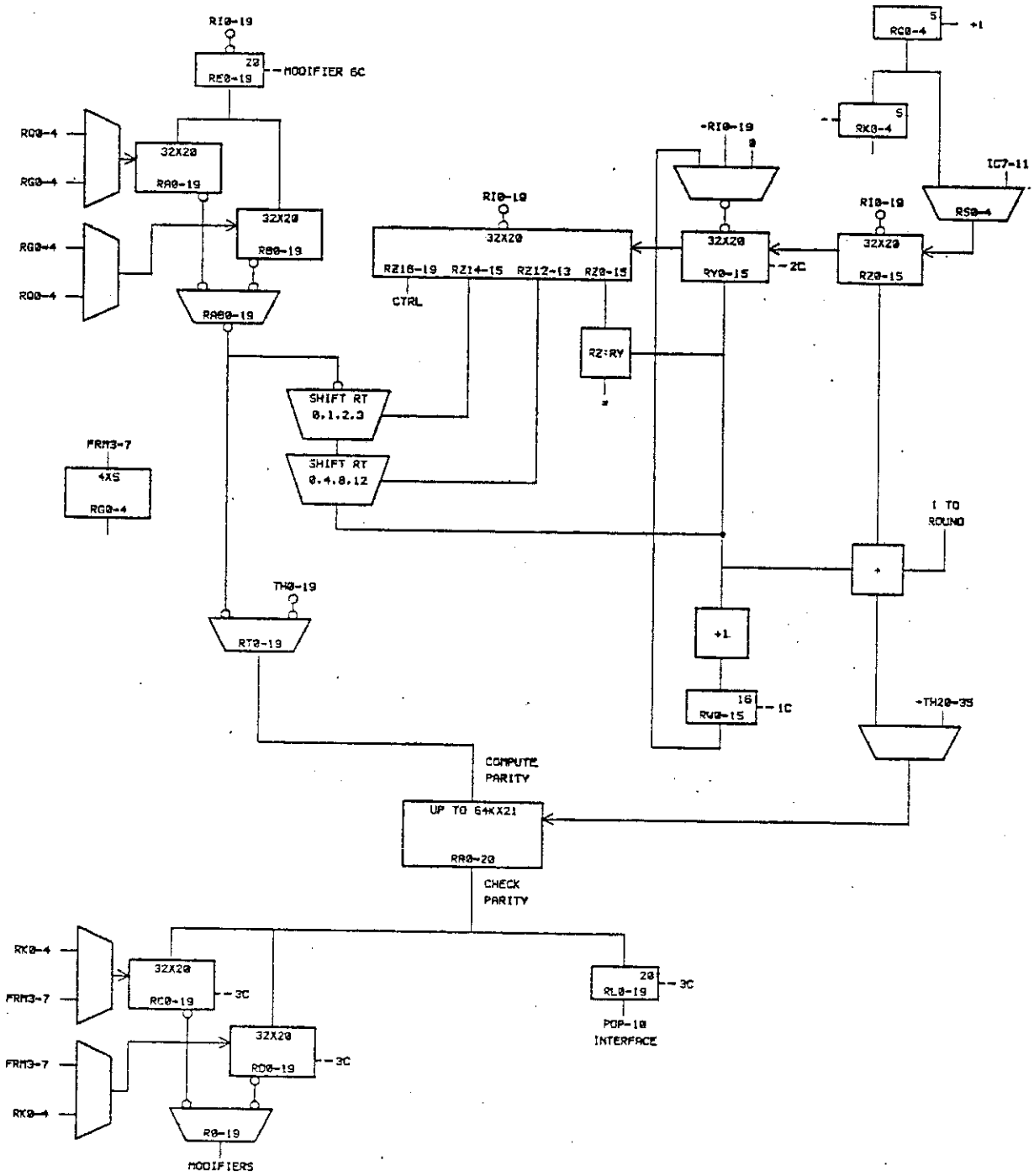



Figure 6. Delay Unit Data Paths

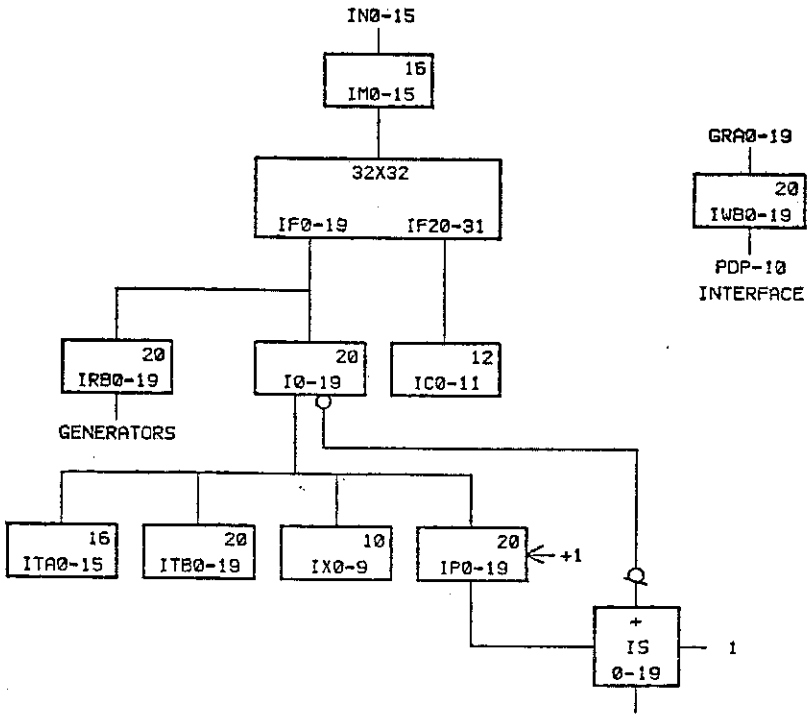



Figure 7. Generic Interface Data Paths

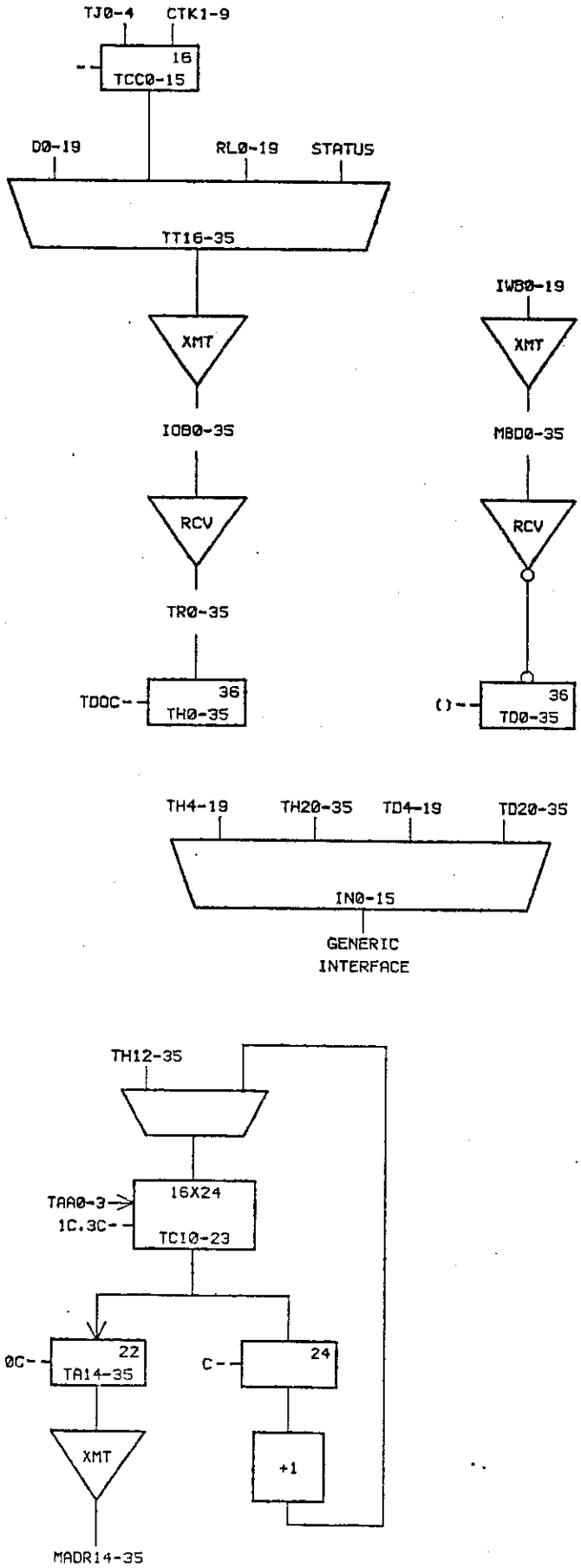



Figure 8. PDP-10 Interface Data Paths