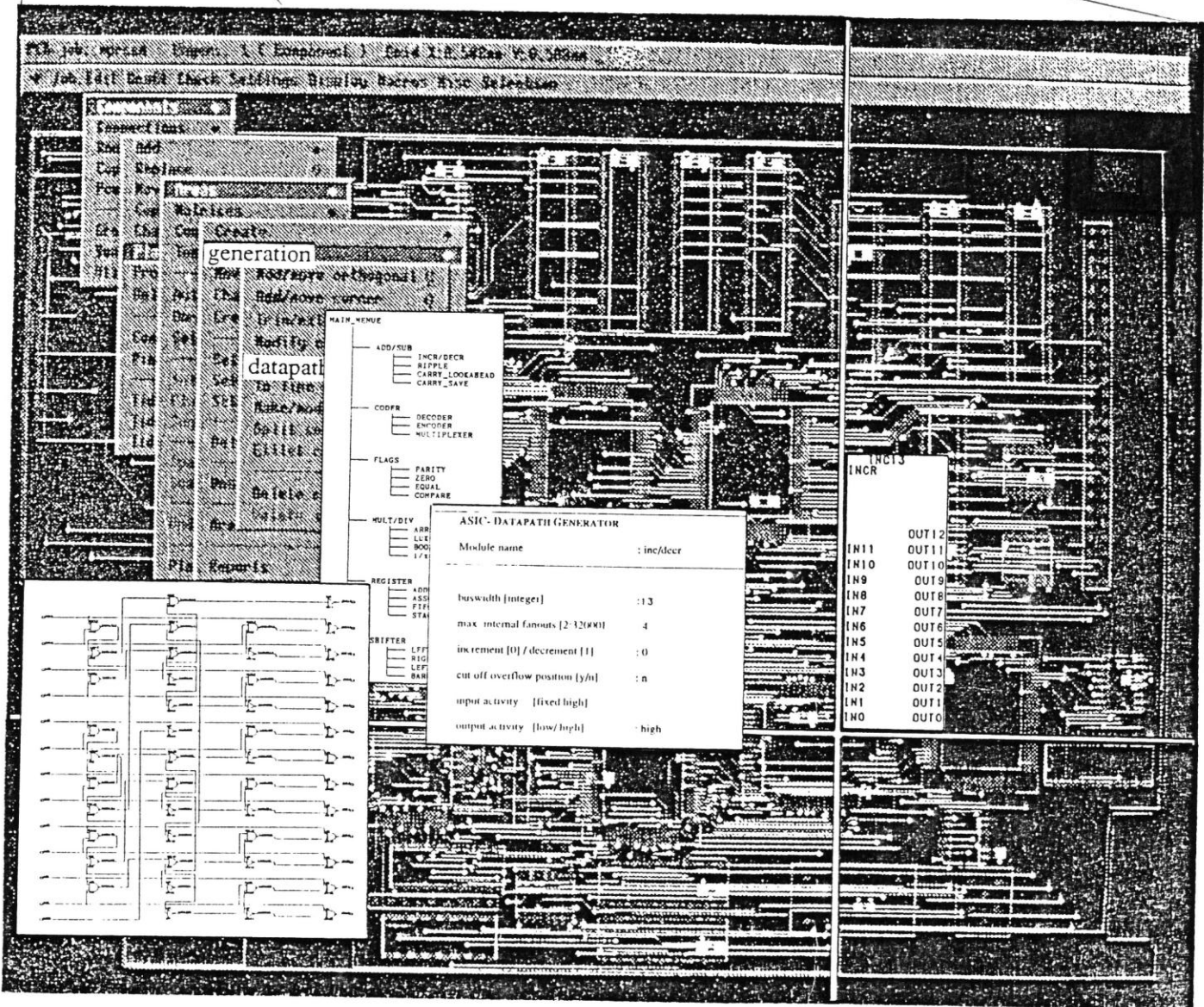
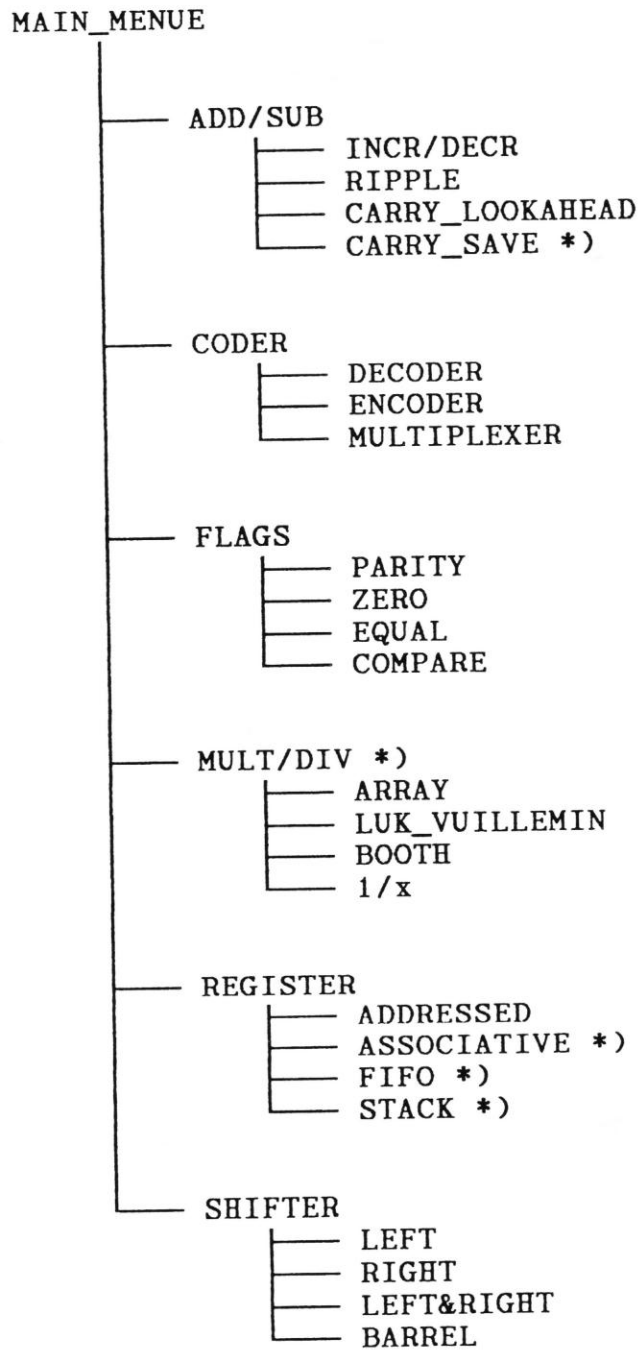


3. DESIGN EXAMPLE



MAIN MENU



(*not available)

GENERATOR MENU

GENERATION EXAMPLE: INCREMENTER 16 BIT

MODULE GENERATOR MENU

ASIC- DATAPATH GENERATOR

Module name : inc/decr

buswidth [integer] : 16

max. internal fanouts [2:32000] : 4

increment [0] / decrement [1] : 0

cut off overflow position [y/n] : n

input activity [fixed high]

output activity [low/ high] : high

GENERATED NETLIST

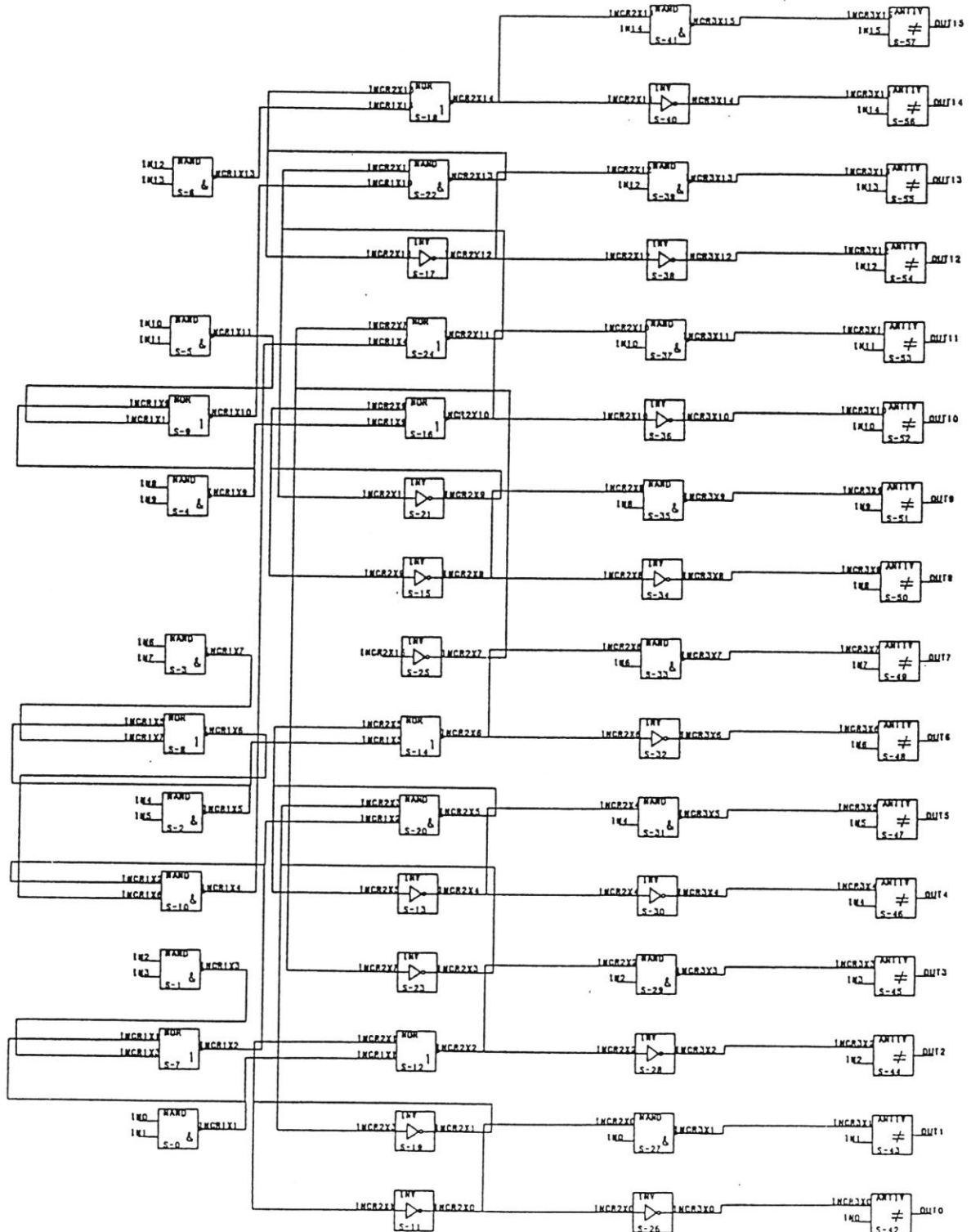
```

! 16,4,0,0,1
!
! _____ increment _____
!
! data input activity: high
! data output activity: high
! data bus width: 16 bit
!
s: test_incl6 #
      e:in(0:15),incr #
      a:out(0:15)
!
ic_1_(1)  in(0)-in(1)-incr1(1)  =nand
ic_1_(3)  in(2)-in(3)-incr1(3)  =nand
ic_1_(5)  in(4)-in(5)-incr1(5)  =nand
ic_1_(7)  in(6)-in(7)-incr1(7)  =nand
ic_1_(9)  in(8)-in(9)-incr1(9)  =nand
ic_1_(11) in(10)-in(11)-incr1(11) =nand
ic_1_(13) in(12)-in(13)-incr1(13) =nand
ic_1_(2)  incr1(1)-incr1(3)-incr1(2) =nor
ic_1_(6)  incr1(5)-incr1(7)-incr1(6) =nor
ic_1_(10) incr1(9)-incr1(11)-incr1(10) =nor
ic_1_(4)  incr1(2)-incr1(6)-incr1(4) =nand
ic_2_(0)  incr2(1)-incr2(0) =inv
ic_2_(2)  incr2(1)-incr1(1)-incr2(2) =nor
ic_2_(4)  incr2(5)-incr2(4) =inv
ic_2_(6)  incr2(5)-incr1(5)-incr2(6) =nor
ic_2_(8)  incr2(9)-incr2(8) =inv
ic_2_(10) incr2(9)-incr1(9)-incr2(10) =nor
ic_2_(12) incr2(13)-incr2(12) =inv
ic_2_(14) incr2(13)-incr1(13)-incr2(14) =nor
ic_2_(1)  incr2(3)-incr2(1) =inv
ic_2_(5)  incr2(3)-incr1(2)-incr2(5) =nand
ic_2_(9)  incr2(11)-incr2(9) =inv
ic_2_(13) incr2(11)-incr1(10)-incr2(13) =nand
ic_2_(3)  incr2(7)-incr2(3) =inv
ic_2_(11) incr2(7)-incr1(4)-incr2(11) =nor
ic_2_(7)  incr-incr2(7) =inv
ic_3_(0)  incr2(0)-incr3(0) =inv
ic_3_(1)  incr2(0)-in(0)-incr3(1) =nand
.
.
.
ic_4_(6)  incr3(6)-in(6)-out(6) =antiv
ic_4_(7)  incr3(7)-in(7)-out(7) =antiv
ic_4_(8)  incr3(8)-in(8)-out(8) =antiv
ic_4_(9)  incr3(9)-in(9)-out(9) =antiv
ic_4_(10) incr3(10)-in(10)-out(10) =antiv
ic_4_(11) incr3(11)-in(11)-out(11) =antiv
ic_4_(12) incr3(12)-in(12)-out(12) =antiv
ic_4_(13) incr3(13)-in(13)-out(13) =antiv
ic_4_(14) incr3(14)-in(14)-out(14) =antiv
ic_4_(15) incr3(15)-in(15)-out(15) =antiv
##

```

GENERATED SCHEMATICS

INCREMENTER 16 BIT



VMS HELP FILES (*.HLB *.HLP)

3 C_LOOKAHEAD

Schneller Adder mit paralleler Uebertragungsberechnung. Der Generator arbeitet mit modifiziertem Wallace-Tree, dh. mit baumartig entklammerten Propagate-Ausdruecken. Damit werden unabhaengig von der Busbreite ausgewogene Fanouts bei fast maximaler Geschwindigkeit erreicht.

4 OVERFLOW

5 POSITIVE

Ueber-/Unterlauf positiver Binaerzahlen (ADD):

00...000 --> 11...111

Zwei n-te Stellen der Summanden ergeben die Moeglichkeit des Ueberlaufes zur (n+1)-ten Stelle im Ergebnis. Der Datenausgang (n+1) stellt den Ueberlauf-Ausgang dar. Eine Subtraktion ist in dieser Zahldarstellung nicht moeglich.

5 TWOS_COMP

Ueber-/Unterlauf bei Zweierkomplementdarstellung (ADD/SUB):

10...000 --> 11...111, 00...000 --> 01...111

Niedersignifikantes Teilwort ohne Vorzeicheninformation:
Zwei n-te Stellen der Summanden ergeben die Moeglichkeit des Ueberlaufes zur (n+1)-ten Stelle im Ergebnis. Der Datenausgang (n+1) stellt in dieser Zahldarstellung den Ueberlaufausgang (CARRY_OUT) dar, der an den Uebertragseingang (CARRY_IN) im naechsthoeheren Teilwort anzuschliessen ist.

Vorzeichen-signifikantes Teilwort:

Die n-te Stelle der Summanden beinhaltet die Vorzeicheninformation.

Ein Ueberlauf e kann auf folgende Arten aus den Summanden

x,y den Uebertraegen c bzw. den Summen s bestimmt werden:

(Busindex in Klammern; Antivalenz: @, AND: *, OR: +, NEG: /)

$$1. e = c(n) @ c(n-1)$$

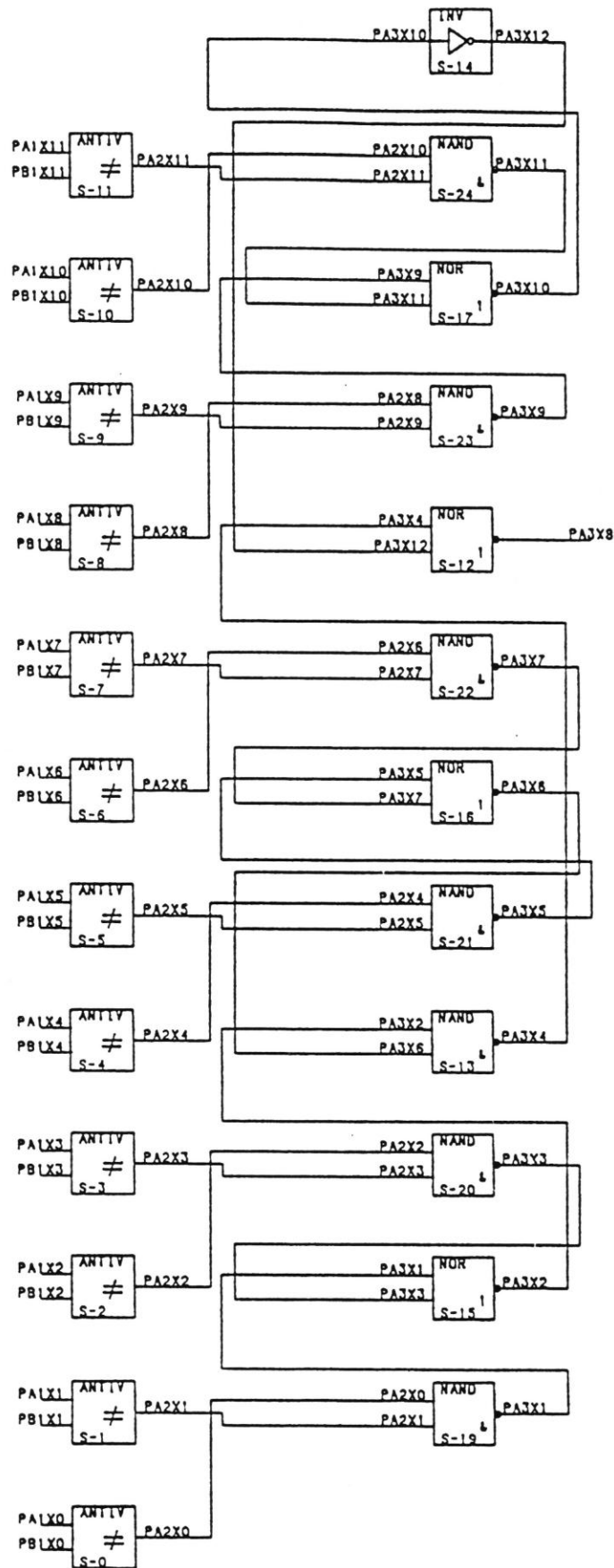
$$2. e = /x(n) * /y(n) * s(n) + x(n) * y(n) * /s(n)$$

Erstere Methode setzt die Zugreifbarkeit auf den Ueberlauf der Stelle (n-1) voraus. Der Ueberlauf c(n) ist identisch der Summe s(n+1), da x- und y- Summanden nur den maximalen Index n besitzen.

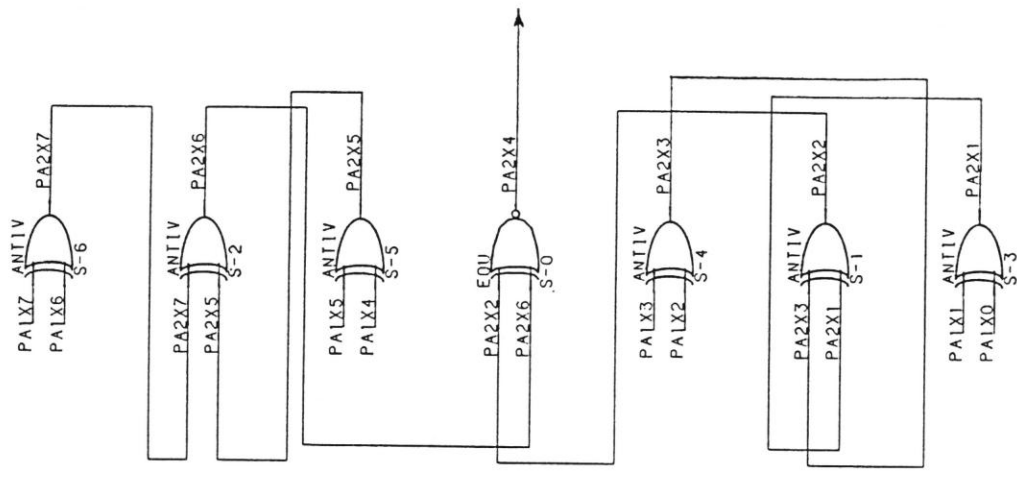
4 PINS

Busbelegung bei Busbreite m Bit:

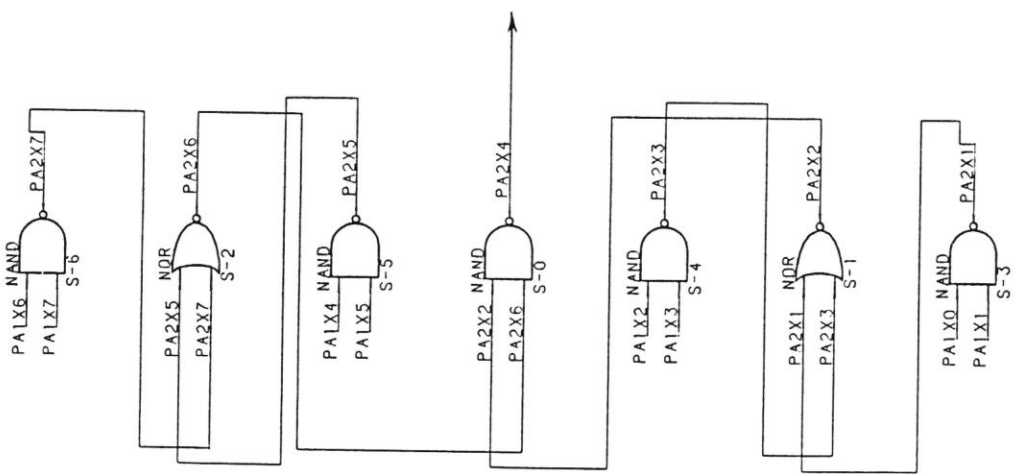
Signal	Anzahl	Name
Carry_input	1	cy
Summand A	m	in1(0...m-1)
Summand B	m	in2(0...m-1)
Ergebnism+1	out	(0...m)



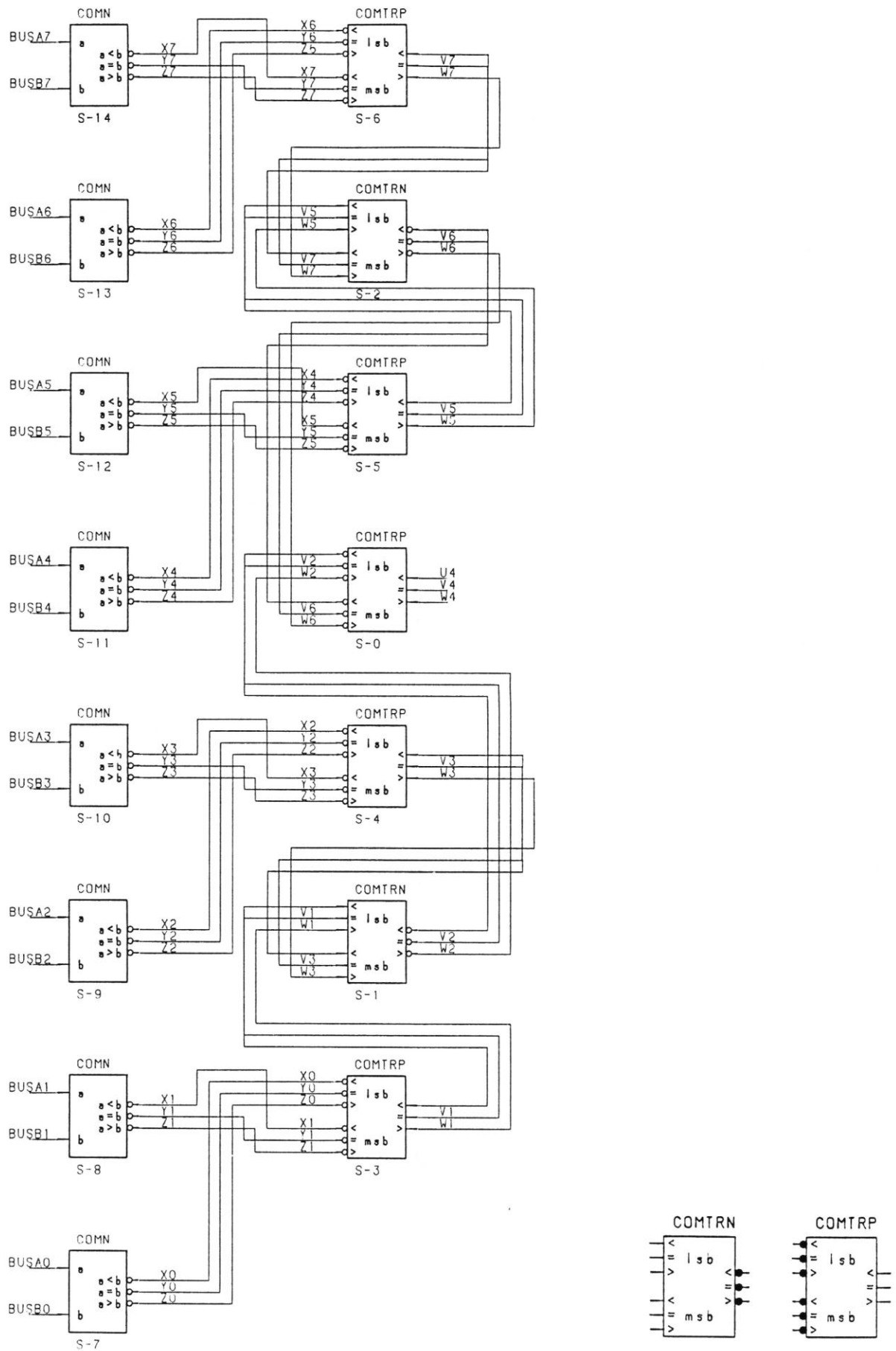
DATA PATH- GENERATION: EQUALFLAG 12 BIT



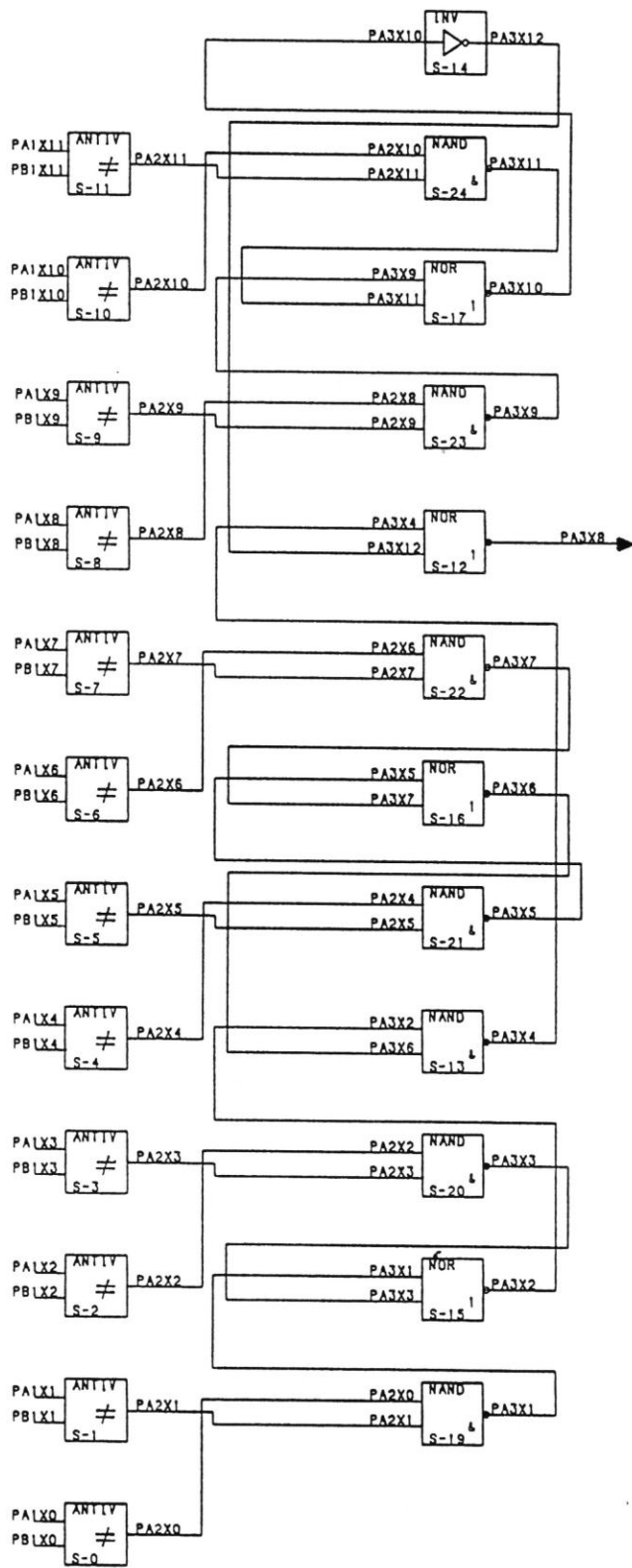
ZEROFLAG 8 BIT



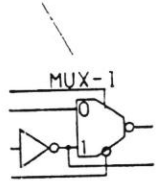
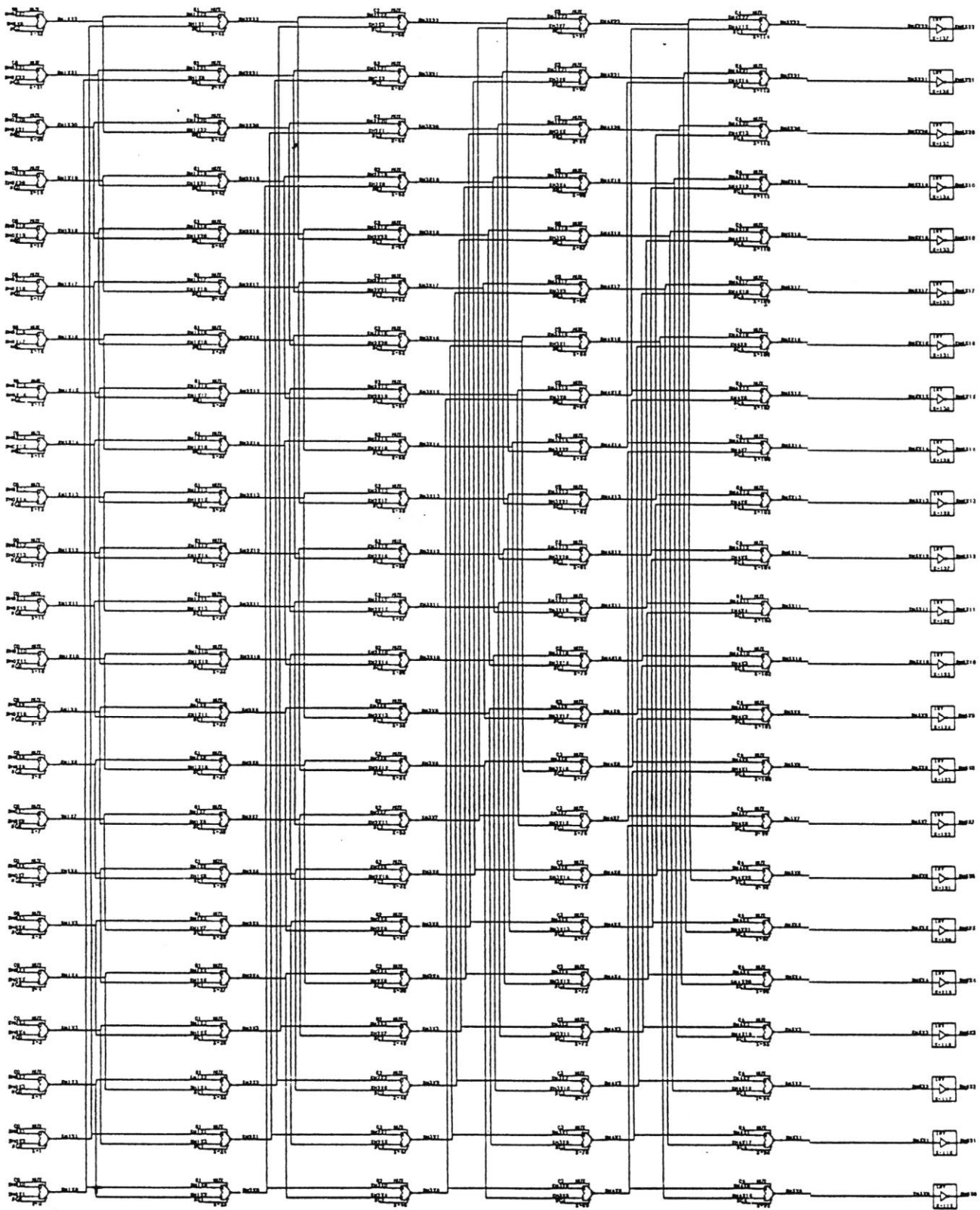
PARITY 8 BIT



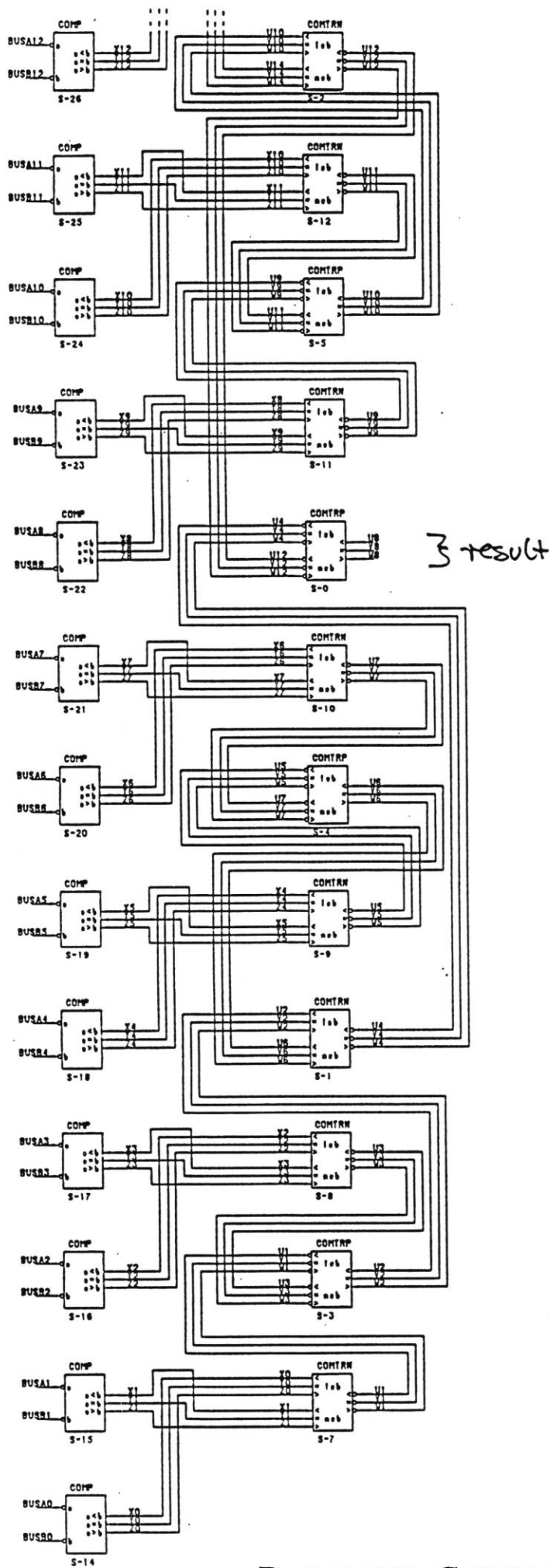
DATAPATH- GENERATION: COMPARATOR 8 BIT



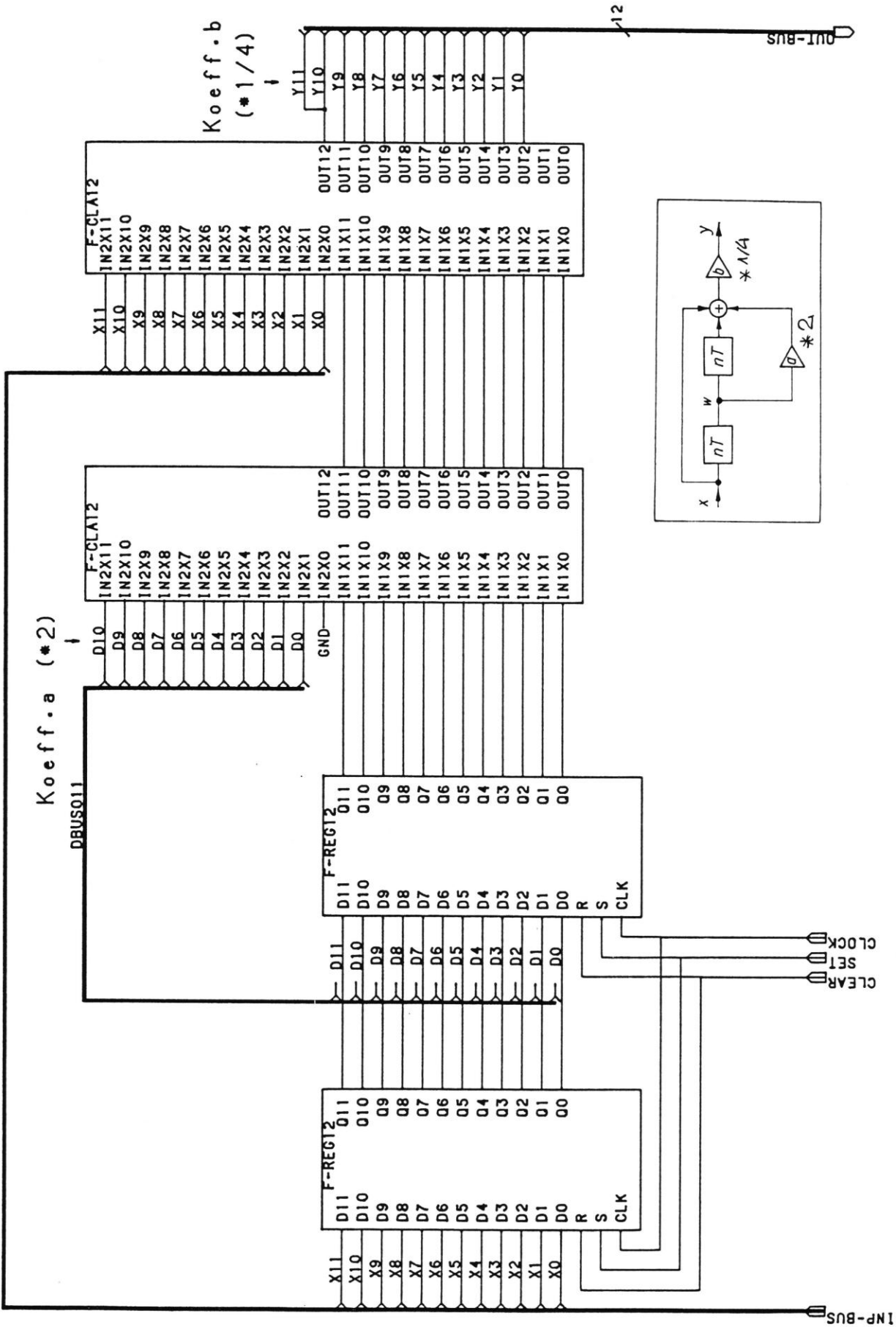
DATAPATH- GENERATION: EQUALFLAG 12 BIT



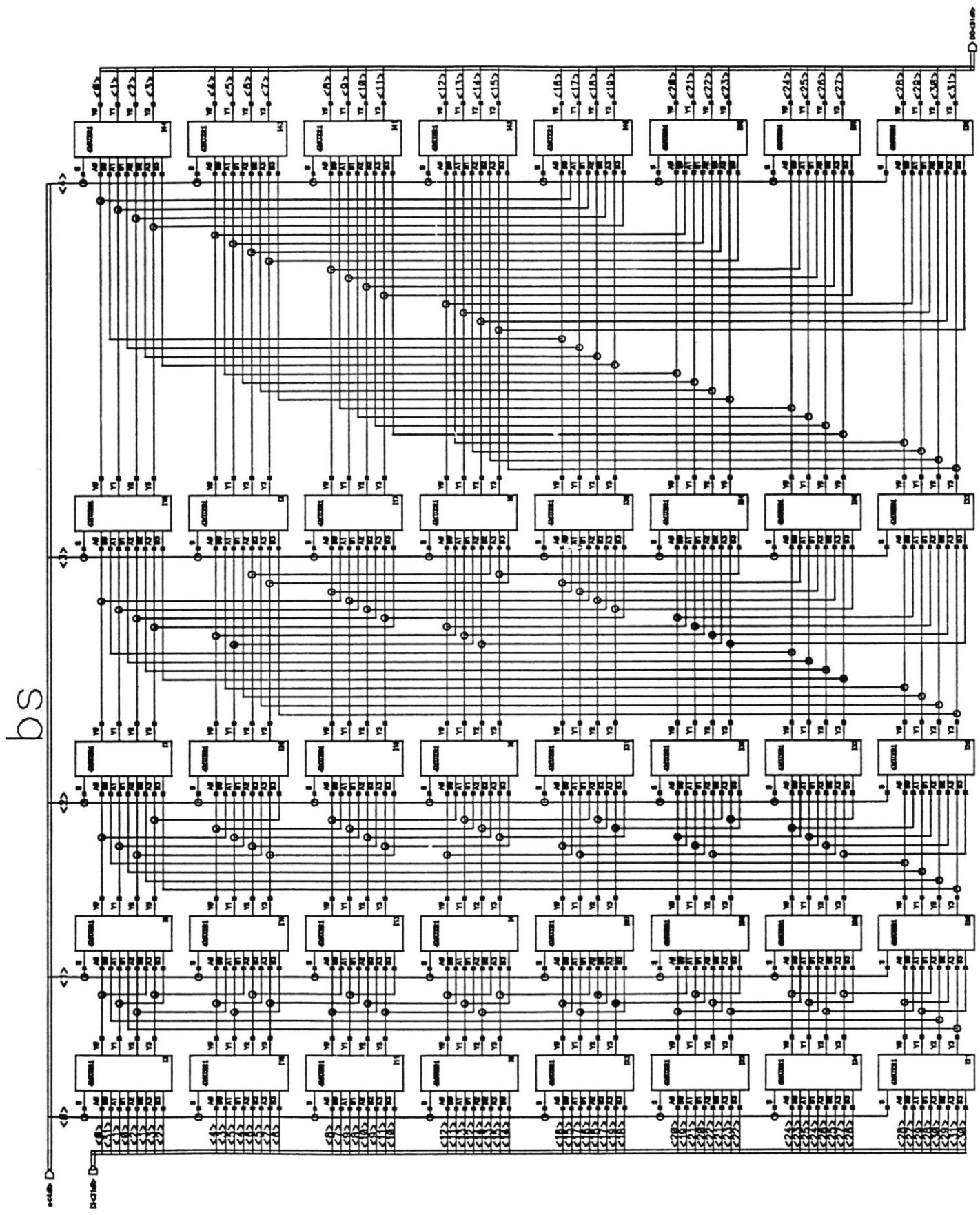
DATAPATH- GENERATION: BARRELSHIFTER 23 BIT



DATAPATH- GENERATION: COMPARATOR 13 BIT



APPLICATION: HIGH SPEED- LOW PASS FILTER 12 BIT



BARRELSHIFTER 32 BIT