A VLSI Implementation of The Collision Avoidance Switch Protocol for CAMB Tree LANs

Viraphol Chaiyakul
Hung Huang
Tatsuya Suda
Daniel D. Gajski

Technical Report #91-14
February 14, 1991

Dept. of Information and Computer Science
University of California, Irvine
Irvine, CA 92717
(714) 856-8059

viraphol@ics.uci.edu

Abstract

To solve a performance bottle neck in random access LANs due to packet collisions and their resolution, collision avoidance switches are introduced. These switches allow random access protocols to achieve high performance by resolving collisions among packets. A conventional hardware implementation of these switches is the use of TTL chips. In this implementation, a handful of TTL chips are required to form a single switch (e.g., 18 TTL chips are needed for an implementation of the CAMB switch [7]). Thus, implementation of a complete network, which requires several of these switches, could very well result in a large and complex hardware system.

Today's modern chip technology allows us to pack large quantity of logic in a single chip. By transferring the conventional implementation of the collision avoidance switches into a VLSI chip, the complexity of the resultant hardware is greatly reduced, not to mention the improvement in hardware performance and ease of packaging.

This report provides an overall study of the collision avoidance protocols for the tree LANs with emphasis on the implementation of collision avoidance switches. Hardware implementations of some of these switches are discussed. And a VLSI implementation of the CAMB switch protocol is introduced.

Contents

1 Introduction 6

2 Broadcast Star Network 7

3 Collision Avoidance Single Broadcast (CASB) Tree 9

4 Tinker Tree 11

5 Collision Avoidance Multiple Broadcast (CAMB) Tree 14

5.1 CAMB Switch Architecture 14

5.2 Station Protocol 15

5.3 Concurrency : Effects of Climbing Packets 16

5.4 Suitability of CAMB tree networks for High-Speed/Optical Networking 18

5.5 Performance of CAMB tree networks 19

6 Implementation of The CAMB Switch in VLSI Technology 19

6.1 Specification 19

6.2 Data Format 21

6.3 Addressing Scheme 23

6.4 Implementation Approach 25

6.5 Implementation 26

6.6 Layout 31

6.7 Performance of the Layout 31

6.8 Signals' Descriptions 32

6.9 Testing 33

7 Conclusion and Future work 40

8 Acknowledgements 40

9 References 41

10 Appendix 42

List of Figures

1 Broadcast Star Switch Architecture 8

2 CASB Tree Network and Switch Architecture. 10

3 Tinker Tree Network. 12

4 Possible Tinker Switch Implementation. 13

5 CAMB Switch Architecture 14

6 Partitioning of CAMB Tree by Climbing Packet. 17

7 CAMB Switch Interface 20

8 Data Format 21

9 Addressing Scheme 23

10 Implementation Hierarchy 25

11 Uplink Selector's Implementation Hierarchy 26

12 Address Recognizer's Implementation Hierarchy 28

13 Downlink Selector's Implementation Hierarchy 30

14 Signals' Descriptions Table 33

15 External I/O Interface of CAMB Switch 43

16 CAMB Switch Schematic Diagram 44

17 Uplink Selector Schematic Diagram 45

18 Mux (SN153) Schematic Diagram 46

19 Priority Resolver Schematic Diagram 47

20 Priority Schematic Diagram 48

21 Binary Counter Schematic Diagram 49

22 Start Recognizer Schematic Diagram 50

23 DFF Pos-Edge With Set Schematic Diagram 51

24 SN74LS153 Schematic Diagram 52
25 Synchronizer Schematic Diagram 53
26 Address Recognizer Schematic Diagram 54
27 End Recognizer Schematic Diagram 55
28 Reset Controller Schematic Diagram 56
29 AR-shift Register Schematic Diagram 57
30 Shift Register Schematic Diagram 58
31 Routing Logic Schematic Diagram 59
32 Sync Schematic Diagram 60
33 DFF Pos-Edge-Triggered With Reset Schematic Diagram 61
34 Comparator Schematic Diagram 62
35 Downlink Selector Schematic Diagram 63
36 Down-Synchronizer Schematic Diagram 64
37 Selector Schematic Diagram 65
38 Layout of the CAMB Switch 66
39 Test Waveform For Binary Counter 67
40 Test Waveform For Comparator 68
41 Test Waveform For Shift Register 69
42 Test Waveform for SN74LS153 (MUX) 70
43 Test Waveform For Priority 71
44 Test Waveform For Uplink Selector 72
45 Test Waveform For Address Recognizer 73
46 Test Waveform for Downlink Selector 74
47 Test Waveform for Test Case 1 75
48 Test Waveform for Test Case 2 76

49 Test Waveform for Test Case 3 77

50 Test Waveform for Test Case 4 78

51 Test Waveform for Test Case 5 79

52 Test Waveform for Test Case 6 80

53 Test Waveform for Test Case 7 81

54 Test Waveform for Test Case 8 82

1 Introduction

Multiple access protocols can be classified broadly into two classes: random/contention access protocols and controlled access protocols. The main difference between the two is their key approach to determine who gets to use the channel when there is competition for it. The random access protocols allow stations to send packets as soon as they are available. If collisions occur, the sender retransmits packets until they are successfully received by the destination stations. Whereas in controlled access protocols collisions are avoided by using polling mechanism coordinating access of the stations to the channel.

Random access protocols exhibit small packet delays under light traffic conditions, while controlled access protocols are efficient when traffic is heavy. The most attractive aspects of the random access protocols are their simplicity, and distributive nature. Networks that use random access protocols appear to be more robust because there is less or no exposure to the failure of a few master nodes.

Regardless of the random access protocols' attractive features, these protocols face a performance bottleneck under heavy traffic conditions due to packets collisions. Most random access protocols handle collision by using some channel capacity to perform transmissions scheduling among contending stations. But a better approach which does not have to sacrificed the channel capacity is to use collision avoidance switches. These switches arbitrate random access to a shared communication channel. While the channel is being used by one station, other stations are blocked from using it hence prevent the collisions. Networks based on these switches can use simple access protocols to provide low packet delay under light load, and they do not waste channel utilization due to collision resolution and the

transmission of collided packets.

In this paper we present summary of some tree LANs architecture that utilize collision avoidance switches protocols: broadcast star (Section 2), Collision Avoidance Single Broadcast (CASB) trees (Section 3), Tinker Tree (Section 4), and Collision Avoidance Multiple Broadcast Tree (CAMB) trees (Section 5). The CAMB tree architecture is proposed in [3]. This tree architecture allows concurrent broadcasts within non-intersecting subtrees. The CAMB tree combines the benefits of random access (low delay when traffic is light; simple, distributed, and therefore robust protocols) with concurrency of transmission, high throughput, and excellent network utilization [5]. More detail discussion of this CAMB tree LANs architecture is given in Section 5. Section 6 introduces an implementation of the CAMB tree switch in VLSI technology. This is accompanied by detail description of implementation approach, hardware schematics, layout, testing, and performance analysis.

2 Broadcast Star Network

This is the simplest kind of the collision avoidance switch protocols. In this network architecture [6], stations are directly connected by full duplex channels to a central switch, thus resulting in the star topology. The switch may be functionally viewed as containing two components: the selector and the broadcaster as shown in Figure 1.

In a normal operation, the selector selects one packet from the Uplinks and transmits this packet on a single output line to the broadcaster. Upon receiving the packet, the broadcaster broadcasts it to all stations through Downlinks. When the selection component

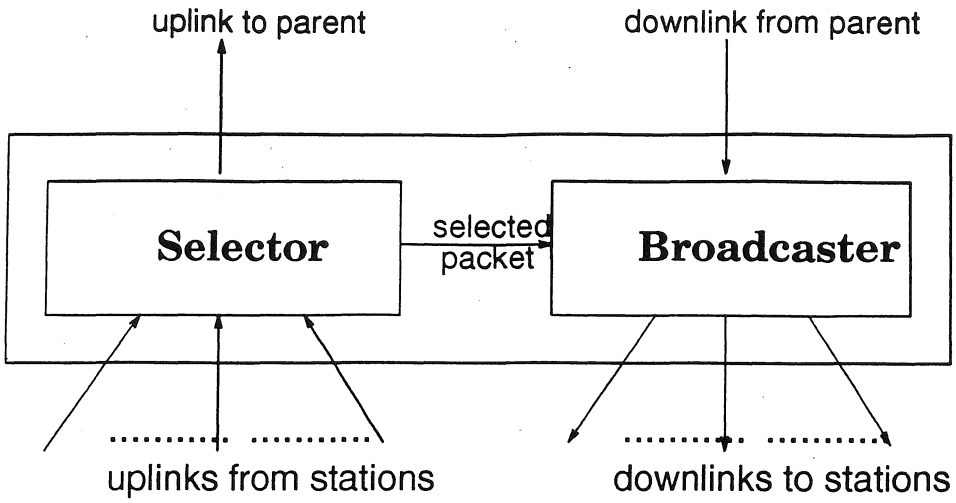

Figure 1: Broadcast Star Switch Architecture

is busy, arriving packets via Uplinks are simply discarded. Thus when two or more packets contend for the output line, it is guaranteed that one of the packets acquires the line and is successfully transmitted on it. As the result, no channel time is wasted in the transmission of collided packets, and the traditional penalty of random access is eliminated.

The station protocol for the broadcast star network architecture can be briefly summarized as follow [4]:

- A station transmits a packet as soon as one is available.
- After a propagation delay to and from the switch, the station monitors its Downlink for the start of its packet.
- If the station does not see the start of its packet, then it retransmits the packet immediately,
- else the station does see its packet and knows that the packet has won the switch and

will be broadcast in its entirety.

The Broadcast Start Network does not require minimum size for the packet length. Furthermore, communication to and from the switch is over point-to-point links. Thus its architecture is suitable for the use in the domain of high-speed optical networking. This is discussed in more detail in [5].

3 Collision Avoidance Single Broadcast (CASB) Tree

A CASB tree is similar to a Broadcast Star but has a general rooted tree topology, as shown in Figure 2a. The nodes of CASB tree consist of switches each of which contains a selector and a broadcaster, as shown in Figure 2b.

The switch for CASB tree architecture is very similar to the Broadcast Star switch, except there is no link between selector and broadcaster components within the switch. Instead the selector passes its output to its parent switch, and the broadcaster receives its input from its parent switch.

Stations in the CASB network use the same protocol as in the simple Broadcast Star network. However, the packet climbs the tree by being selected by each switch along the packet's path to the root. The root switch serves the same function as the central switch in the Broadcast Star. Any packets selected by the root's selector is broadcast to its children. These children repeat the broadcast to their children and so forth until the broadcast reaches every station. Thus this produces distributed effect of the packet section over several levels of switches.

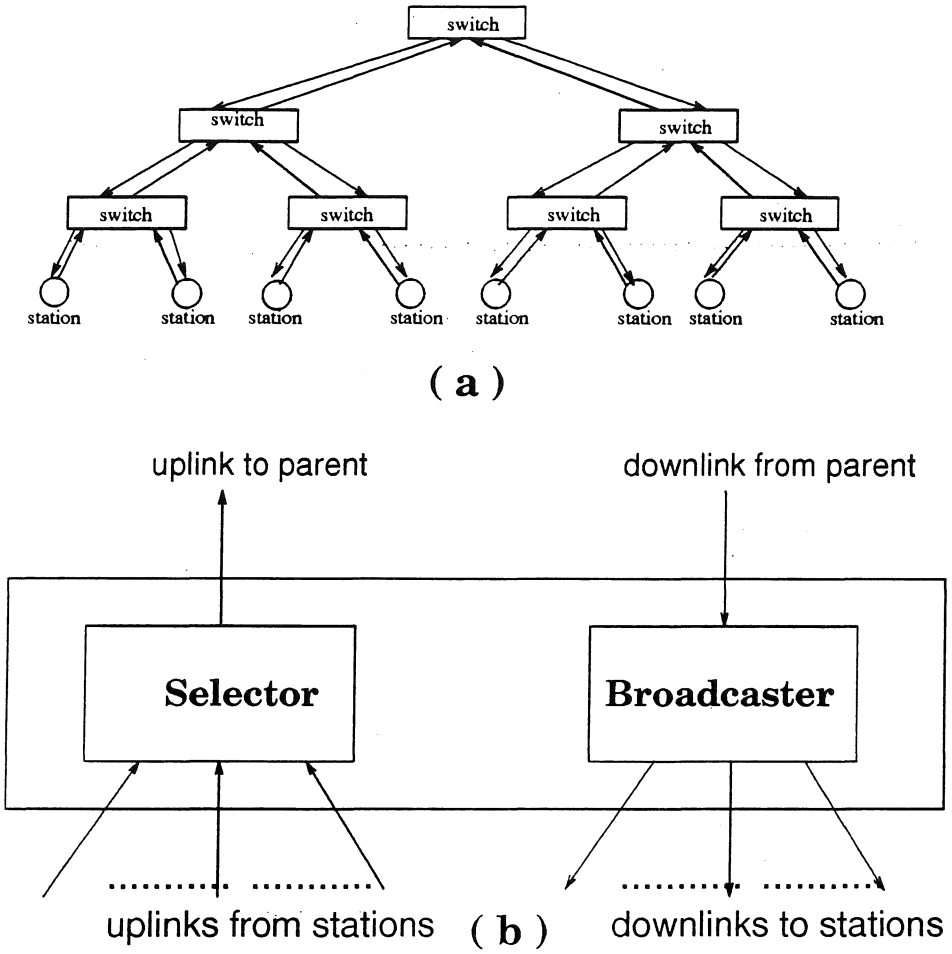

Figure 2: CASB Tree Network and Switch Architecture.

Three main advantages of CASB tree over the simple broadcast star network architecture are:

- The CASB tree allows easy expansion when the fanout of a switch is fully occupied. Because all switches in the network are identical, expansion simply requires connection of a new switch as a child of the full switch.
- The CASB tree provides some tolerance to the vulnerability of the network switches. This is because of distributive nature of the CASB tree.
- The cabling cost for CASB tree is much less excessive than the Broadcast tree. This is because CASB tree allows cables to be shared instead of running a separate cable for each station to a central switch.

4 Tinker Tree

Tinker Tree is a point-to-point network which uses collision avoidance and random access protocols. The switches are connected to each other using full duplex connections in a tree-like structure. Packets are routed by the switches along the unique path using address information contained in packet headers. Tinker Tree allows concurrent packet transmissions if the transmission paths do not overlapped. Contention along overlapping path segments is resolved by the collision avoidance circuitry in the switches. Basically, when contention occurs, only one packet is allowed to access the transmission line and others are blocked and need retransmission by their source stations. This is illustrate in Figure 3.

Two possible architecture of Tinker tree's switches are introduced in [4]. One of which

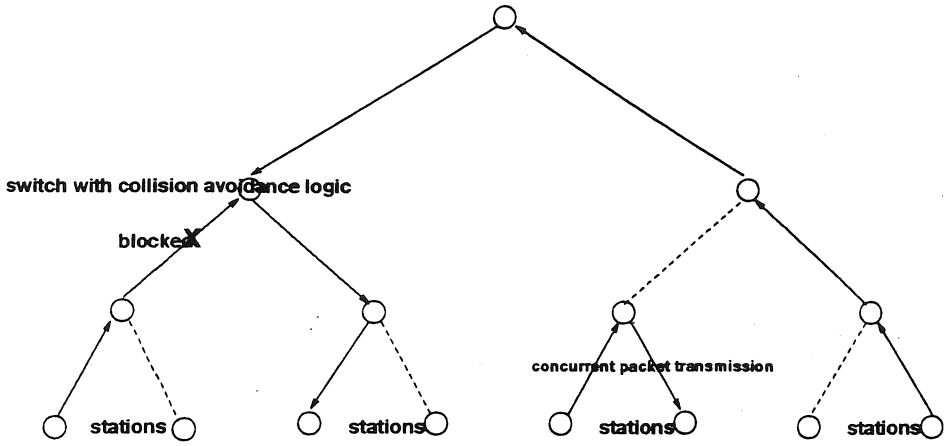

Figure 3: Tinker Tree Network.

is shown in Figure 4a.

In this architecture, the switch consists of four components: Uplink Selector, Router, Downlink Selector, and the second Router. The Uplink Selector is identical to the Uplink Selector of the CASB switch. The Router responsible for selection of packets to either the parent Uplink or the Downlink selector. The Downlink Selector functions exactly like the Uplink Selector by providing collision avoidance among its two inputs. Lastly, the second Router obtains its input from the Downlink Selector, and routes the packet to one of the child Downlinks according to the addressing information in the packet's header. This architecture supports at most two simultaneous transmissions, as illustrates in Figure 4b.

Since Tinker tree is not a broadcast in nature, detection of transmission success and failure requires extra mechanism. One method, which is suggested in [4], that could be used to detect blocked packets, is to have the destination station sends a short acknowledge packet to the source station if the packet has been successfully transmitted. In addition, the source station learns of transmission failure by timing out on the receipt of an acknowledgement.

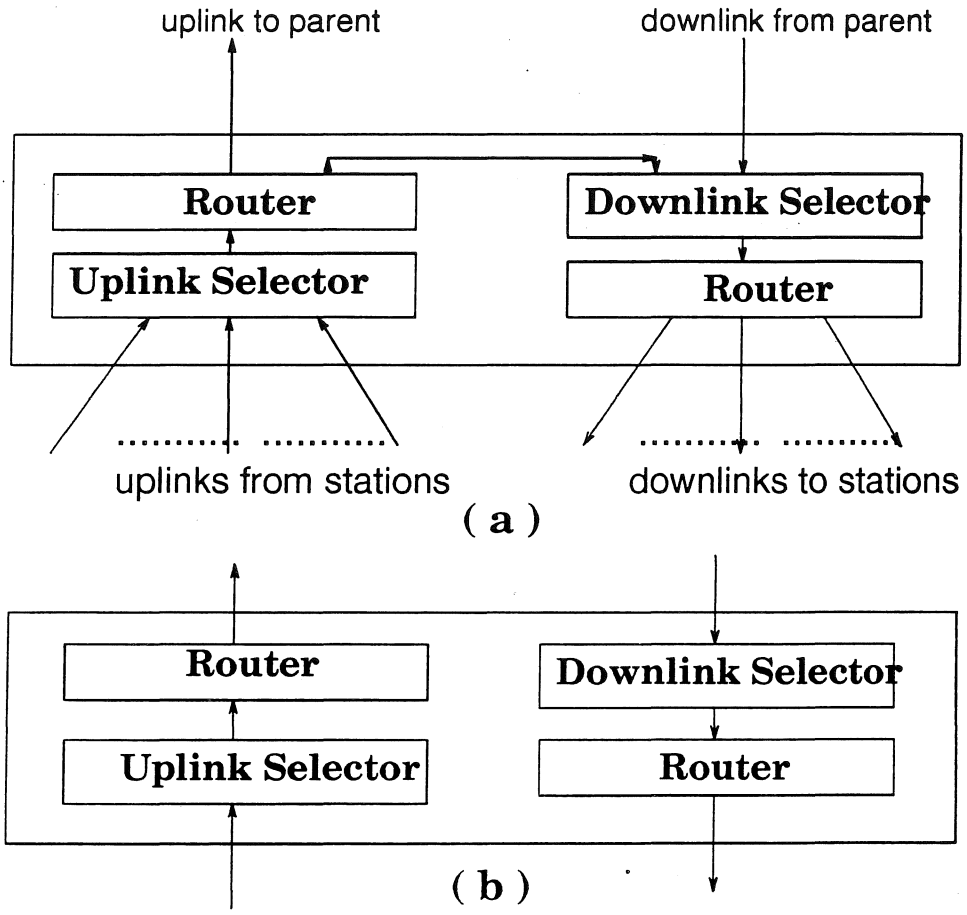

Figure 4: Possible Tinker Switch Implementation.

5 Collision Avoidance Multiple Broadcast (CAMB) Tree

The CAMB tree architecture is proposed in [3]. The CAMB tree is a CASB tree with modified switch that allows broadcast to the subtree below any switches. A packet reaches its destination by climbing the tree and being broadcast by its proper ancestor to the subtree below. The proper ancestor of a packet is the switch that roots the minimal subtree containing both the source and destination stations of the packet.

5.1 CAMB Switch Architecture

The CAMB switch has three components: the Uplink Selector (US), the Address Recognizer (AR), and the Downlink Selector (DS), as shown in Figure 5.

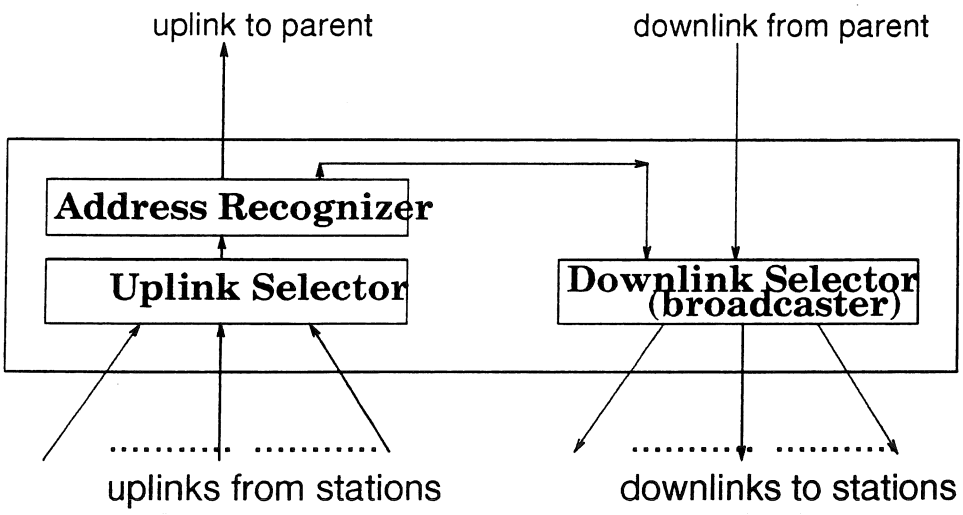

Figure 5: CAMB Switch Architecture

The US implements collision avoidance among packets arriving on children Uplinks. The

first packet arriving on an Uplink is accepted and its bit stream is passed to the AR. Any packets arrives at the US while it is busy are blocked. The AR determines whether or not to broadcast the packets. If the switch is not the proper ancestor of the packet, the AR transmits the packet to the parent Uplink. Otherwise, if the switch is the proper ancestor of the packet then AR chooses to broadcast the packet by passing the packet to DS. However, if DS is busy at that time, the AR discards the packet it is receiving (broadcast preemption). Else the DS is idle and AR transmits the packet to the DS. The AR also transmits a copy a copy of packet to the parent Uplink if the switch is the proper ancestor. This is to keep the US of a switch above the packet's proper ancestor busies. Thus prevent stations beneath that switch from using the switch as a broadcast point. This makes the packet's own broadcast less vulnerable to abortion.

The DS receives packets from either the AR or the parent Downlink. Packets received from the parent Downlink are given priority over those received from the AR. Otherwise it would be possible for a source station to receive the broadcast of its packet without the broadcast being received by the destination station. If the DS is busy broadcasting a packet from the AR when packet from the parent Downlink arrives, the AR-broadcast is terminated and packet from the parent Downlink is broadcast instead (broadcast abortion).

5.2 Station Protocol

The CAMB tree's station protocol is based upon the station monitoring its Downlink for the broadcast of its packet. Because of the possibilities of broadcast abortion and preemption, the CAMB tree's station protocol requires that a station see the broadcast of its entire

packet. The CAMB tree's station protocol can be briefly summarized as follow:

A station transmits a packet as soon as one is ready, starting at say, time t. The station monitors the Downlink for the packet broadcast. Packet is retransmitted if the station does not see the start of its packet by the time $t + R_{pa}$ (where R_{pa} = round trip propagation delay between the station and the proper ancestor. This value can be determined during network initialization). Packet is considered successfully transmitted only if the station sees the broadcast of the whole packet, since broadcast abortion is possible.

5.3 Concurrency : Effects of Climbing Packets

Concurrency is made possible by the point-to-point, segmented nature of the network which allows broadcasting to subsets of the tree. As pointed out earlier, packet at its proper ancestor will be transmitted by the AR to both DS and the parent Uplink. Every time a packet climbs above its proper ancestor, it busies the US of the switch above its proper ancestor. As the result, it partitions the tree into broadcast domains. Within each of these partitions, broadcast may occur. However, any attempt to transmit from within a partition to a destination outside the partition will not succeed. This is illustrates in Figure 6.

In the Figure 6, each indicated transmission is possible except for the one marked with the cross (because the parent of its partition is busy).

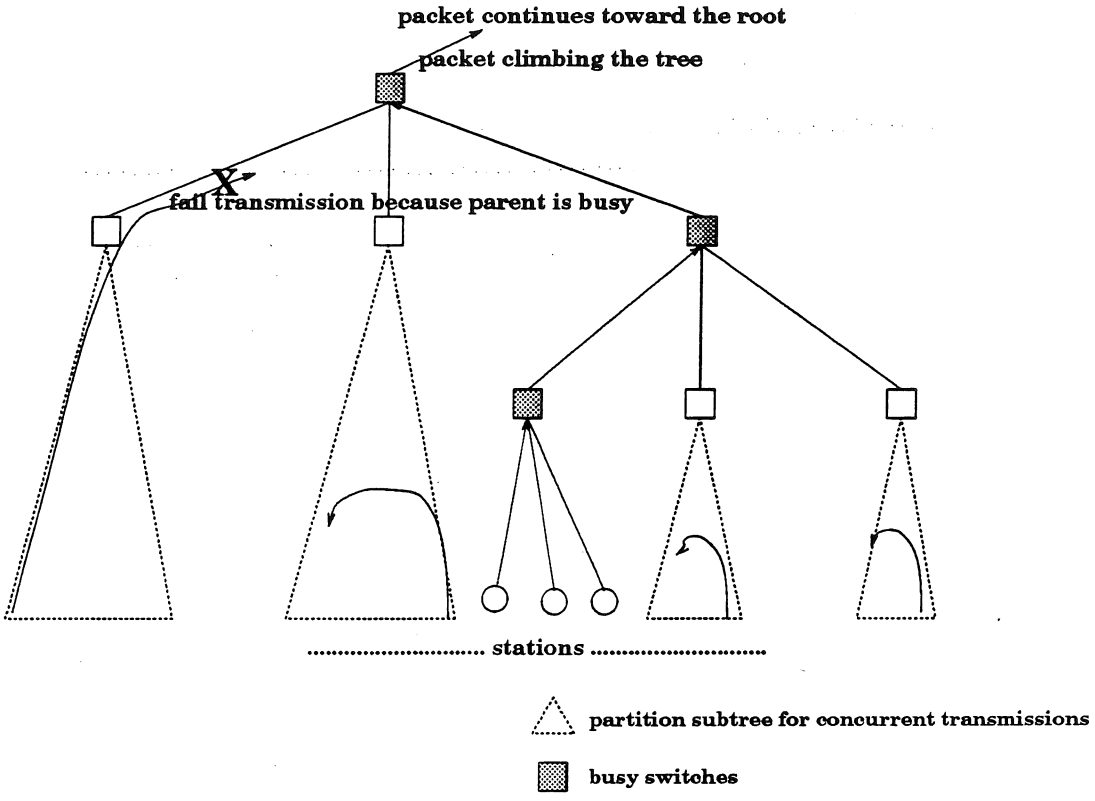

Figure 6: Partitioning of CAMB Tree by Climbing Packet.

5.4 Suitability of CAMB tree networks for High-Speed/Optical Networking

Some aspects of the CAMB tree network which suggests its suitability for use in the high speed, optical networking are summarized as follow :

- The CAMB tree networks provide point-to-point physical architecture which is compatible to fiber optics technology.
- The switch protocol can be implemented using current photonic device technology because of its simplicity. Furthermore, the switch does not store and forward packets; the memory required by the switch is small and can be of the form of FIFO storage. This allows the possibility of using fiber optic delay lines as a storage mechanism. Detail discussion of the photonic implementation of CAMB tree switch can be found in [5].
- There is no minimum packet duration requirement in the CAMB tree network. Hence, doubling the channel speed does not violate any protocol requirements in CAMB tree architecture.
- The use of high channel speed in the CAMB tree reduces the probability of broadcast preemption and abortion, and thus reduces the average transmission time of the packets.

5.5 Performance of CAMB tree networks

In the preformance study of CAMB tree given in [5], it is shown that the CAMB tree network gives better performance than the broadcast star network. Furthermore, increasing the channel speed of the network to the optical range of 1 Gbit/sec dramatically reduces the transmission delay in the CAMB tree networks.

6 Implementation of The CAMB Switch in VLSI Technology

6.1 Specification

This section provides a specific implementation of the CAMB switch in the form of high level description of specification [7].

- Switch Interface

  The following is a list of specific interface requirements for this implementation of the CAMB switch, Figure 7.

  - Each switch has four Uplinks and four Downlinks. This means leaf switches can interface with four devices (stations). Thus, the network formed by these switches will appear like a 4-ary tree.
  - Each switch is connected to its parent switch with one Uplink and one Downlink.

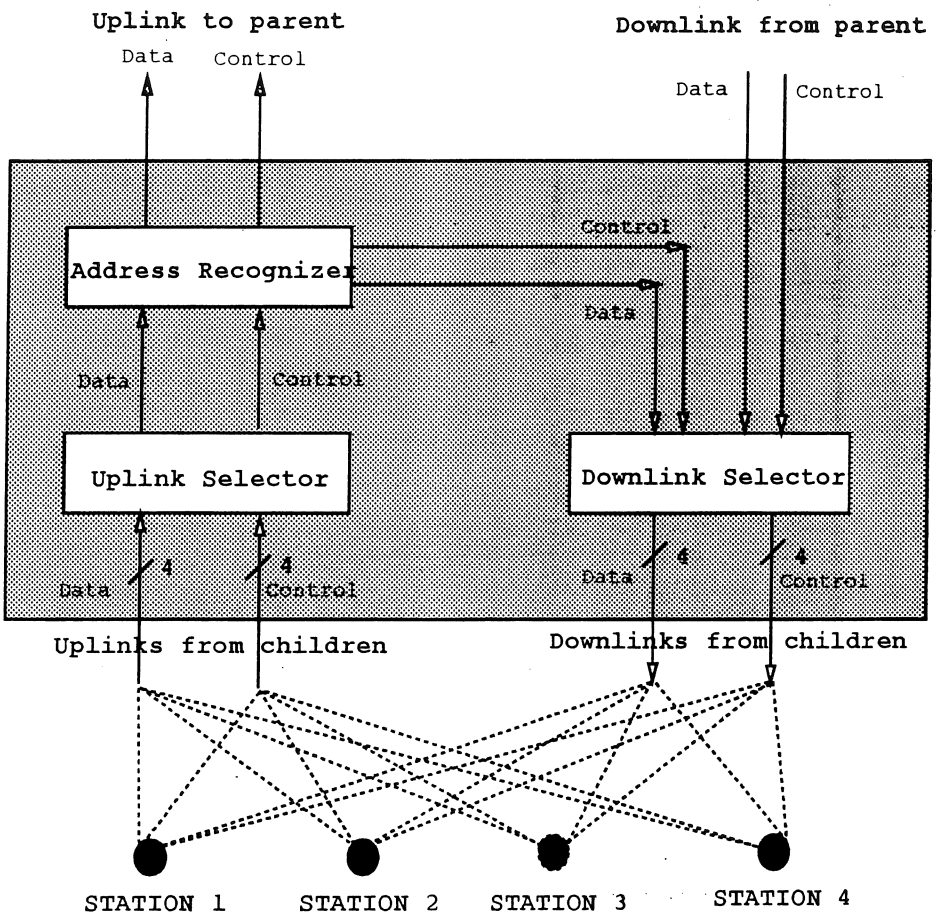

Figure 7: CAMB Switch Interface

– Each Up/Downlink consists of two lines: Data line and Control line. The Data line is used to transmit data from switch to switch. Whereas the Control line acts as the indicator for beginning and ending of packet transmission.

6.2 Data Format

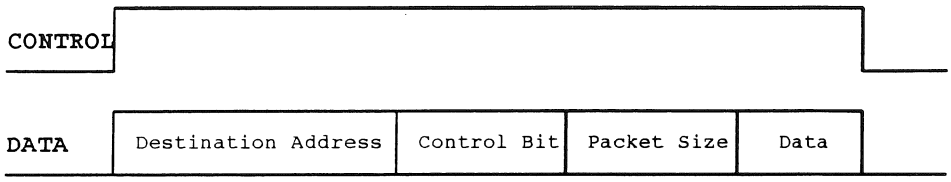

Figure 8: Data Format

- **Control Format (Control line)**

The control line for this implementation of the CAMB tree LANs is used as the transmission controller (Figure 8). This signal tells the beginning and ending of the transmission. The beginning of transmission is indicated by the rising edge of the control signal. The control signal remains high as long as there is a valid packet transmission in the data line. The falling edge of the control signal indicates the ending of the transmission.

- **Packet Format (Data line)**

The packet format for this implementation of the CAMB tree LANs comprises of four fields (Figure 8): the Destination Address Field, the Control Bit Field, the Packet Size Field and the Data Field. Out all of these four fields, only the first two are used by the switch.

- *The Destination Address Field* indicates the destination address of the packet. This field is assigned to be 8 bits long.

- *The Control Bit Field* is 1 bit long field. Packet that reached its proper ancestor will also be sent up to busy its parent nodes, the climbing packet. However, this climbing packet should not be broadcasted again. Thus, this field is used to indicate if the packet has reached its proper ancestor. This bit is initially set to 1 by the sending station. It remains 1 until the packet reaches the proper ancestor. At its proper ancestor switch, this bit is reset to 0, and, from then, it acts as a climbing packet. At the same time, the packet's proper ancestor also transmits another packet down to its children switches, the broadcast packet. And to ensure that the broadcast packet correspond to its check-sum and other coded fault detection values, the Control Bit Field of this broadcast packet is unchanged.

- *The Packet Size Field* indicates number of bits contained in the data field. This information is used by the stations (not by the switches) to determine whether the packet has been aborted. If the Packet Size Field is equal to the size of the data field, the packet is successfully transmitted, otherwise, abortion occured. Note that, depending on higher layer protocols, other overhead information such as source address and CRC may be necessary in the packet.

- *The Data Field* contains transmission message from source to destination devices (stations).

6.3 Addressing Scheme

The addressing scheme assume in this CAMB switch design is illustrated in the Figure 9.

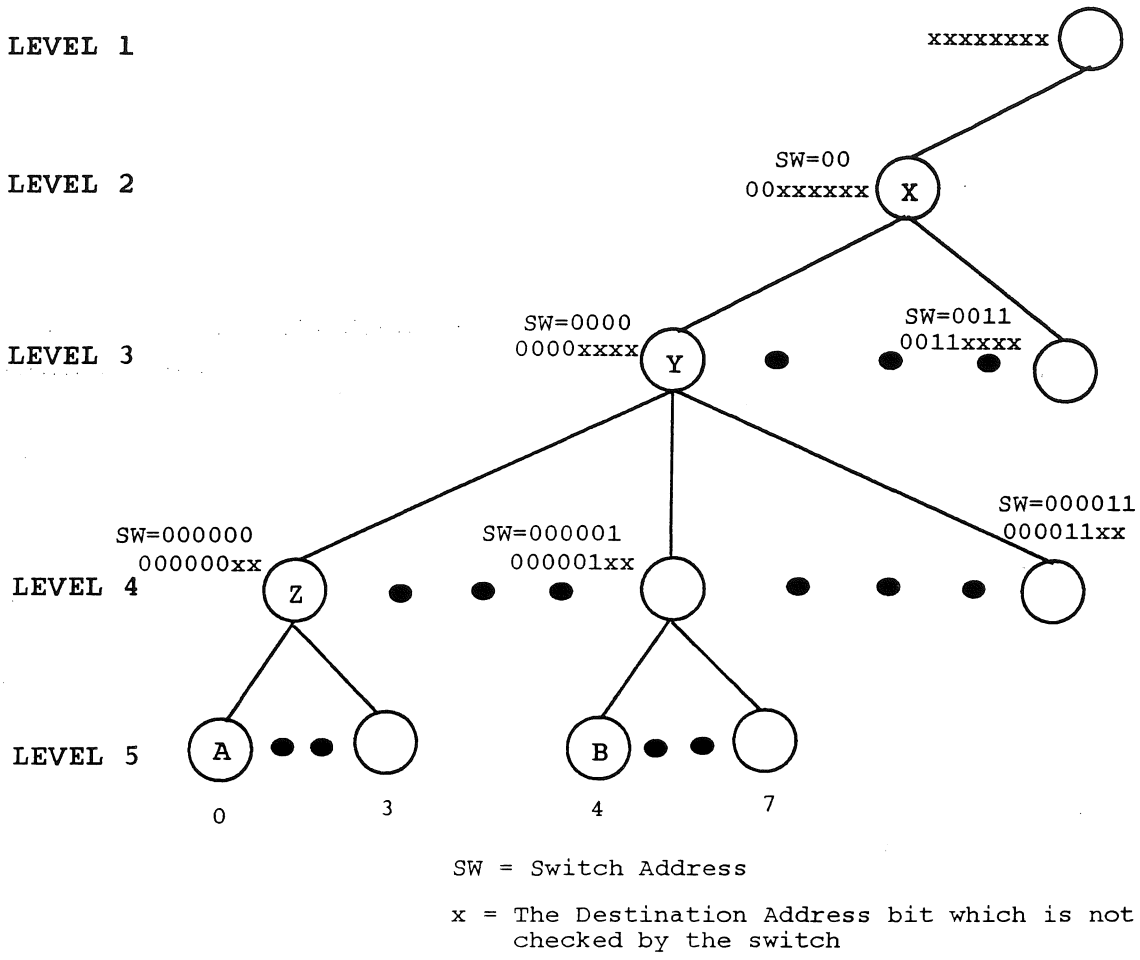

Figure 9: Addressing Scheme

Stations(devices) are ordered in ascending ordered from left to the right of the tree. Each station has a $2 * (M - 1)$ bit long address, where M is the height of the tree. Address are assigned to the switches in the following way. The level i switches have a $2 * (i - 1)$ bit long address. The address of a switch on the level i matches with the first $2 * (i - 1)$ bits of the address of all the station in the subtree.

This stations and switches numbering scheme makes it easy for a switch to determine whether it is the proper ancestor of a packet. A switch on level i checks $2 * (i - 1)$ leftmost bit of the Address Field of a packet. If it is equal to its own address, then it is the proper ancestor of the packet. In this case, the packet is sent to both the parent switch (climbing) and the Downlink switches (broadcast). Otherwise, the packet is only sent upward to the parent switch.

Figure 9 illustrates a transmission of a packet from station A (address 00000000) to station B (address 00000100) as an example. The immediate parent of switch A, switch Z (address 000000), checks the destination of the packet (Destination Address Field). As the six leftmost bits of the destination address (000001) is different from address of the switch Z (000000), Z sends the packet up to its only parent switch Y. Switch Y performs the same packet address checking but only on the four leftmost bits. This time, the switch address (0000) and packet address (0000) match. Realizing that it is the proper ancestor of the packet, switch Y broadcast the packet to its children. The packet is also sent upward (climbing packet) to the parent switch (X). In addition, the Control Bit Field of the climbing packet is reset to 0 by the proper ancestor (Y) to indicate that the packet has already been broadcasted.

At switch X, the two leftmost bits of the destination address field of the packet(00) is compared with the switch address (00). Since both of them are the same, switch X knows that it is the ancestor of the packet. However, if switch X actually broadcasts the packet, the packet will be broadcasted twice. This will tremendously decrease the performance of the network. To avoid this, the switch checks the Control Bit Field of the packet header before broadcasting the packet. The value 0, indicates that the packet has already been broadcasted

by its proper ancestor, so the switch only transmits this packet upward to its parent switch (this is what happens at switch X). Otherwise, the switch broadcasts the packet if it is the proper ancestor.

6.4 Implementation Approach

The implementation of the CAMB switch, from its specification given above, is broken down into number of hierarchical level, as shown in Figure 10. Schematics for each of these parts are given from Figure 15 to Figure 37, in their hierarchical level order.

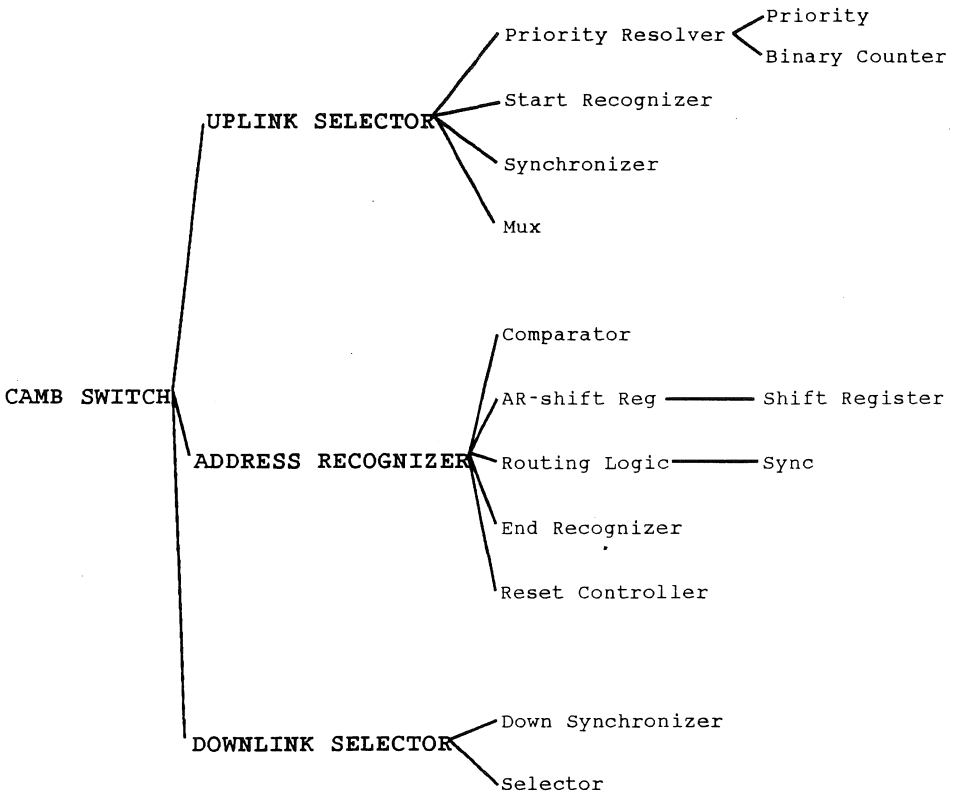

Figure 10: Implementation Hierarchy

The description of first few levels of implementation hierarchy are given in the next

section with the hope that the rest are self explanatory.

6.5 Implementation

The implementation of the switch is broken into three main parts (described in section 5.1): the Uplink Selector, the Address Recognizer and The Downlink Selector.

- **The Uplink Selector**

  The implementation of the Uplink Selector (Figure 11) is further broken into four parts:

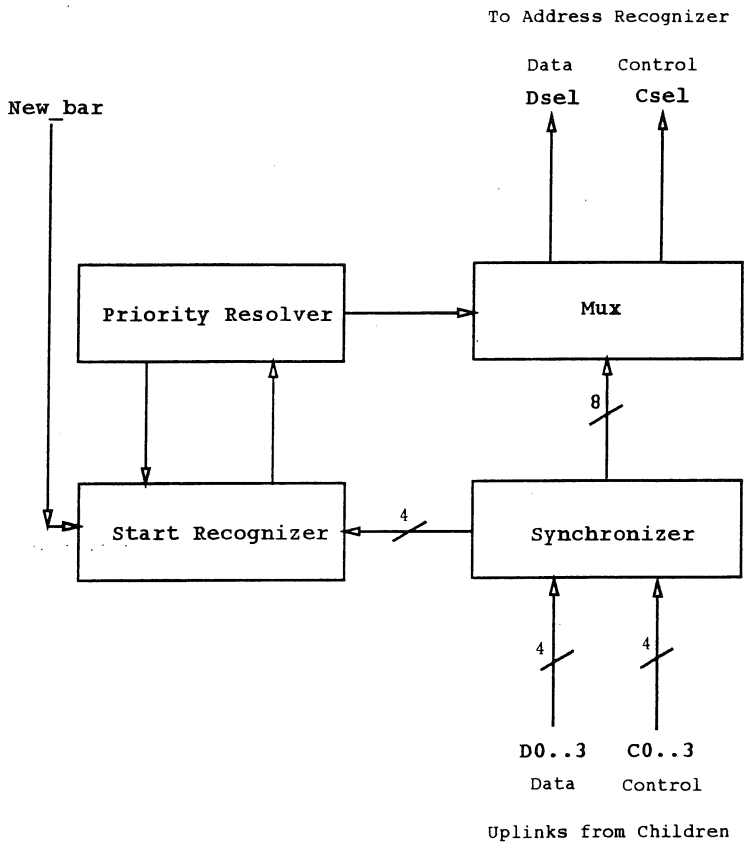

Figure 11: Uplink Selector's Implementation Hierarchy

- *The Synchronizer* responsible for aligning the incoming packet with the switch's internal clock. One important observation: if several switches in the network share the same clock signal, this unit can be removed to speed up the transmission.
- *The Start Recognizer* detects the start of the new packet. This unit also allows the Uplink Selector to block all other Uplinks' transmission while selected packet is being transmitted.
- *The Priority Resolver* selects one packet from the set of to be transmitted packets, at random.
- *The Mux* which passes only the packet selects by the Priority Resolver up to the Address Recognizer.

Schematic diagrams of the Uplink Selector's implementation is given in Figure 17 to Figure 25.

- **The Address Recognizer**

The Address Recognizer (Figure 12) is broken into three main parts:

- *The AR-shift Registers* comprises of three shift registers (Figure 29). The Shift-Register 1 which is responsible for synchronizing of the packet's destination address (the Destination Address Field) so that it can be checked with the switch address by the Comparator. The Shift-Register 1 also responsible for resetting the Control Bit Field of the packet if the switch is the packet's proper ancestor. Shift-Register 2 which is responsible for synchronizing data to be sent up to the parent switch. The Shift-Register 3 which is responsible for synchronizing the control to be sent up to the parent switch.

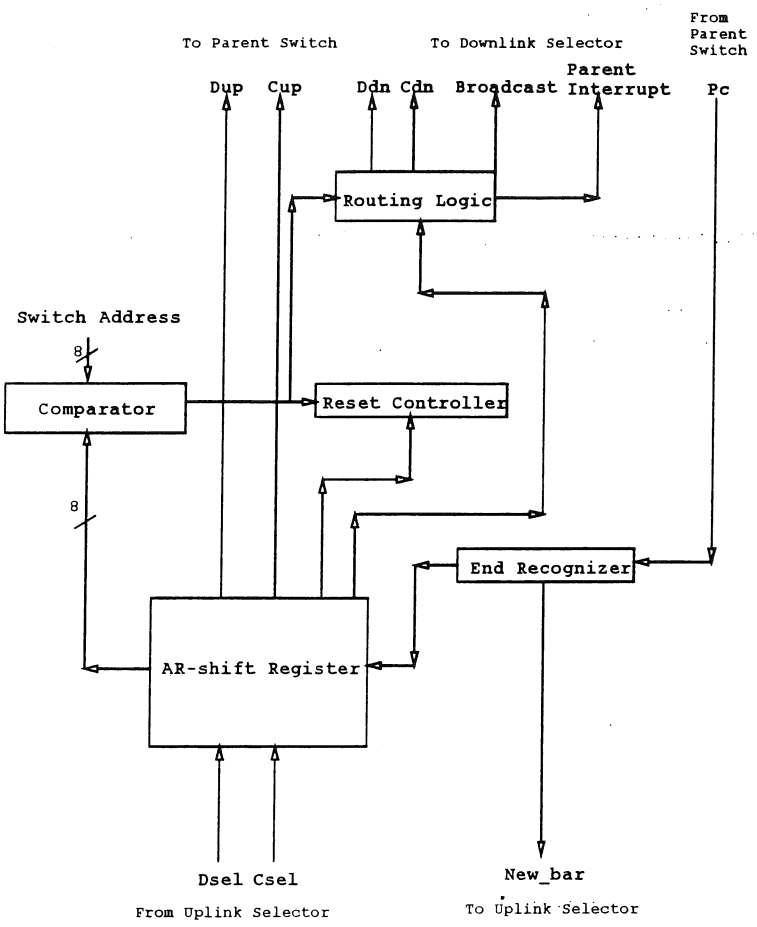

Figure 12: Address Recognizer's Implementation Hierarchy

- *The Comparator* matches the switch address with the packet's destination address.
- *The Routing Logic* generates a signal to indicate whether the in-comming packet should be sent to the Downlink Selector (broadcast) or not.
- *The Reset Controller* generates a signal whenever the switch recognizes that it is the proper ancestor of the packet. This signal is used by the Shift-Register 1 (in the AR-shift Registers) to reset the Control Bit Field of the packet.
- *The End Recognizer* detects the end of a transmission or interrupt from the parent Downlink line (packet abortion). When one of these events happens, the End Recognizer sends a reset signal to the Uplink Selector.

Schematic diagrams of the Address Recognizer's implementation is given in Figure 26 to Figure 34.

- **The Downlink Selector**

The Downlink Selector (Figure 13) composes of two major parts:

- *The Down Synchronizer* is used to force the control line to go low when the packet abortion occurs.
- *The Selector* selects a packet either from the parent Downlink or from the Address Recognizer to broadcast. Higher priority is given to the parent Downlink's packet. So, if a packet from parent Downlink arrives during a packet transmission from the Address Recognizer, the selector will abort the broadcast and transmits the parent Downlink's packet instead (packet abortion). In addition, any packet from Address Recognizer will be blocked during the parent Downlink transmission (packet preemption).

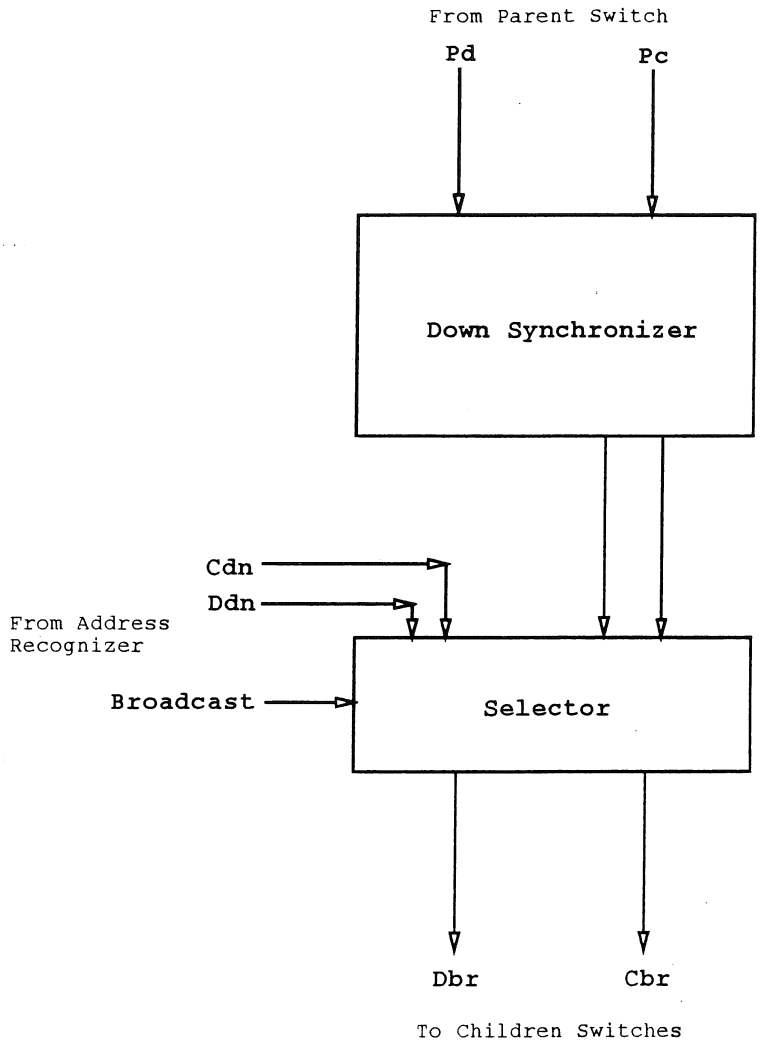

Figure 13: Downlink Selector's Implementation Hierarchy

Schematic diagrams of the Downlink Selector's implementation is given in Figure 35 to Figure 37.

6.6 Layout

To obtain layout for the CAMB switch, a standard cell layout synthesis tool, Autocells[9] (GDT[9] Version 4.0.1g), is used. Autocells is an automatic place and route tool for laying out circuits using standard cells (polycells). Standard cells are small, predefined rectangular layout blocks that perform simple logic functions that correspond to low-level icons in a schematics. This Autocells is provided as a part of the integrated GDT system.

The layout produced by this system is shown in Figure 38. This layout is in 3 microns P-well standard cmos technology. Routing is done in only 1 layer of metal. And to obtain the smallest layout area that can possibly be generated by the Autocells system, all transistors are set to its nominal size.

The resultant layout has the dimension of 1497.5 (width) by 1549.5 (height) microns. Input and output ports of the switch are arranged as shown in Figure 15.

6.7 Performance of the Layout

Circuit and layout testing is accomplished with Lsim[9] simulator (GDT version 4.0.1g). Three basic simulation algorithms are supported by the Lsim simulator. These include system or logic simulation with single direction signal flow, bi-directional switch level simulation, and ADEPT circuit and analog level simulation. In this CAMB switch implementation, the

bi-directional switch level simulation is used as the functional testing method. And performance of the layout is obtained by using the Lsim's ADEPT mode simulation.

In one of the test cases, Test Case 7, the switch is driven with the clock speed of 10 Mhz, which is the normal Ethernet's channel speed. The result obtained shows correct functionality of the CAMB switch's layout with input to output delay of 32.78 ns.

6.8 Signals' Descriptions

This section provides descriptions to signals which are used in later sections and schematic diagrams.

Signals	Descriptions
Inputs	
CLKK	A periodic clock. All components used in the design are positive edge triggered.
RESET_BAR	The master switch reset signal. When activated, all latches are cleared, and New_bar signal is activated. This causes abortion of any ongoing transmissions, and put the switch back to its "*ready to receive new packet*" mode.
D0..3	Data inputs. The switch supports up to four children stations/devices
C0..3	Control inputs from children stations/devices.
Pd	Parent switch's downlink data line.
Pc	Parent switch's downlink control line.
Internal signals	
New_bar	When activated, the switch resets to recieve a new packet. This signal is activated if one of the following events occur: 1) end of packet transmission, 2) interrupt causes by transmission from the parent switch, 3) switch reset.
Dsel	Data line for selected transmission.
Csel	Control line for selected transmission.
Broadcast	Indicator for packet broadcast.
Parent_Int_Bar	Indicator for transmission from the parent switch.
Ddn	Data line for Downlink Selector.
Cdn	Control line for Downlink Selector.
Outputs	
Dup	Data line for upward transmission (to parent switch).
Cup	Control line for upward transmission (to parent switch).
Dbr	Data line for broadcast transmission (to children switches).
Cbr	Control line for broadcast transmission (to children switches).

Figure 14: Signals' Descriptions Table

6.9 Testing

Test waveforms for components used in the design are given from Figure 39 to Figure 54. This section gives explanation only to those test waveforms for major components (Figure 44 to Figure 54) with a hope that the rest are self explanatory. Explanation for each test vector is accomplished using connection scenario in Figure 7 and descriptions of signals from Figure 14.

- **Test waveform for the Uplink Selector (Figure 44)**

  This test case is used to determine the correct functionality of the Uplink Selector. Events in the test vector are numbered so that they can be easily refer to, as follow:

  1. Station 0 requests transmission by raising signal C0.
  2. Completion of transmission from station 0.
  3. Station 3 requests transmission by raising signal C3.
  4. Completion of transmission from station 3.
  5. Transmission from station 2 is blocked because the switch is processing station 3's request.
  6. Transmission from station 2 still being blocked because the switch is in the middle of transmission (New_bar signal is low).
  7. Transmission from station 2 is serviced.

- **Test waveform for the Address Recognizer (Figure 45)**

  This test case is used to determine the correct functionality of the Address Recognizer. The address of the switch is set to be 00000010. Events in the test vector are numbered so that they can be easily refer to, as follow:

  1. Request of transmission by rasing the Csel. Since the destination address of the packet (00000010) is equal to the switch address, transmission to both Cup and Cdn is performed.
  2. Transmission of data.
  3. Control Bit Field of the packet (Dsel) is set to high, indicating that the packet has not been broadcasted yet.

4. Upward climbing packet has the Control Bit Field reset to avoid duplicate broadcast.

5. The Control Bit Field of the broadcast packet remains the same(high).

6. Transmission completes.

7. Reset the state, and ready for new packets.

- **Test waveform for the Downlink Selector (Figure 46)**

  This test case is used to determine the correct functionality of the Downlink Selector. Events in the test vector are numbered so that they can be easily refer to, as follow:

  1. Request for broadcast with data in Ddn and control information in Cdn.

  2. Parent switch requests packet transmission. Since, this has a higher priority than the broadcast, Ddn broadcast is aborted (packet abortion).

  3. Transmission of the parent switch data, Pd, begins.

  4. Parent switch transmission completes.

  5. Broadcast starts again.

- **Testing of the CAMB Switch (Figure 47 to Figure 54)**

  The following test cases are used to check the performance of the CAMB switch as a whole. In all test casses, the address of the switch is set to be 00000010.

  - **Test Case 1 (Figure 47)**

    This test vector shows the scenario where station 1 transmits packet to the destination station 00010010. Since the switch is not the proper ancestor of the packet,

it only transmits data upward to its parent switch (through Dup and Cup). This is indicated in the following events:

1. Station 1 requests transmission to station 00010010.
2. Data from station 1 is being transmitted.
3. Transmission completes.

- **Test Case 2 (Figure 48)**

  This test vector shows the scenario where station 2's transmission is blocked because the switch is transmitting station 1's packet. This is indicated in the following events:

  1. Station 1 requests transmission to station 00010011.
  2. Data from station 1 is being transmitted.
  3. Station 2 requests transmission but is blocked because packet from station 1 is being transmitted.
  4. Transmission from station 1 completes.
  5. Station 2's transmission request is accepted. The packet's destination address is 10001110
  6. Transmission from station 2 completes.

- **Test Case 3 (Figure 49)**

  This test vector shows the scenario where station 1 transmits packet to the destination station 00000010. Since the switch is the proper ancestor of the packet, it sends data both upward to its parent switch (through Dup and Cup) and broadcast the packet (through Dbr and Cbr). This is indicated in the following events:

  1. Station 1 requests transmission to station 00000010.

2. Transmission from station 1 is being broadcasted.

3. Data from station 1 is being transmitted.

4. The Control Bit Field of the broadcast data is unchanged.

5. The Control Bit Field of the climbing data is reset to prevent multiple broadcasts.

6. Transmission from station 1 completes.

- **Test Case 4 (Figure 50)**

  This test vector shows the scenario where station 1 transmits a climbing packet (ie., station 1 can be thought of as the proper ancestor switch of the packet), by resetting the Control Bit Field of the packet. Even though this switch is the proper ancestor of the packet, the switch will not broadcast the packet, since the packet is a climbing packet. This is indicated in the following events:

  1. Station 1 requests transmission to station 00000010.

  2. Packet from station 1 is transmitted upward to the parent switch.

  3. The Control Bit Field of the packet is low, hence, no broadcast is performed.

  4. Indication of no broadcast.

  5. Transmission from station 1 completes.

- **Test Case 5 (Figure 51)**

  This test vector shows the scenario where parent switch's transmission is given a higher priority than the broadcast from a child station (packet abortion). The broadcast packet will be dropped once the parent switch requests transmission. This is indicated in the following events:

  1. Station 1 requests transmission to station 00000010.

2. Packet from station 1 is transmitted upward to the parent switch.

3. Packet from station 1 is broadcasted.

4. The Control Bit Field of the packet from station 1 is high. This is why the packet can be broadcasted.

5. The Control Bit Field of the broadcast packet remains the same.

6. The Control Bit Field of the upward climbing packet is reset to prevent multiple broadcasts.

7. Parent switch requests transmission.

8. Since parent switch's transmission has a higher priority, the transmission from station 1 is aborted and transmission from the parent switch takes over.

9. Transmission from parent switch completes.

- **Test Case 6 (Figure 52)**

This test vector shows the scenario where parent switch's transmission is given a higher priority than the broadcast from a child station. It also shows the concurrency in upward transmission to the parent switch and broadcast from the parent switch. This is indicated in the following events:

1. Station 1 requests transmission to station 00000010.

2. Packet from station 1 is both transmitted upward and broadcasted.

3. The Control Bit Field of the packet from station 1 is high. This is why the packet can be broadcasted.

4. The Control Bit Field of the broadcast packet remains the same.

5. The Control Bit Filed of the upward climbing packet is reset to prevent multiple broadcasts.

6. Parent switch requests transmission.

7. Since parent switch's transmission has a higher priority, the transmission from station 1 is aborted and transmission from the parent switch takes over.

8. Upward climbing packet from station 1 is also blocked.

9. Station 2 requests transmission to station 00010110.

10. Packet from station 2 is transmitted upward.

11. Parent switch's transmission completes.

12. Station 2's transmission completes.

- **Test Case 7 (Figure 53)**

  This test vector is the same as the test vector in Test Case 1. However this test case is run in analog simulation with all output terminals (Dbr, Cbr, Dup, and Cup) set to drive a load of 0.7 pf. The clock speed is set to 10 Mhz (which is the common Ethernet's channel speed). The obtained result shows that the switch still function correctly under these conditions.

- **Test Case 8 (Figure 54)**

  The figure for this test case is basically a magnified region of the Test Case 7 (The region is shown in the Figure 53). From this figure, the delay time for the data transmission is obtained. Since all transitions in the circuit happen at the raising edge of the clock, the transmission delay is equivalent to the delay from the raising edge of the clock to the changing of data value. This delay is measured to be 32.78 ns.

7 Conclusion and Future work

In this report, we described and compared several network architectures based on collision avoidance protocols: a Broadcast Star, CASB tree, Tinker Tree, and CAMB Tree.

The CAMB Tree network proposed by [3] has the potential of combining the benefits of random access(low delay when traffic is light; simple, distributed, and therefore robust protocols) with excellent network utilization and concurrency transmission. We described its station protocol and presented an implementation of its switch architecture in VLSI technology. The implemented layout is tested with numerous test data. And, in one of the test case, its performance is shown to be suitable for standard Ethernet system, with 10Mbps channel speed.

As for future work, to obtain the VLSI chip implementation of the CAMB tree switch, implementation of chip Input-Output interface (e.g., I/O pad, and drivers) is required. Furthermore, possibility of packing several of CAMB switches in a single VLSI chip should be explored.

8 Acknowledgements

The authors would like to thank Allen Wu for his advice and suggestions.

9 References

[1] K. Goto, and T. Suda, Performance Analysis of a Broadcast Star Local Area Network with Collision Avoidance: Part1, Infinite Station Population Model, Submitted to the IEE International Conf. on Distributed Computing Systems, 1990.

[2] K. Goto, and T. Suda, Performance Analysis of a Broadcast Star Local Area Network with Collision Avoidance: Part2, Finite Station Population Model, Submitted to the IEE International Conf. on Distributed Computing Systems, 1990.

[3] T. Suda, S. Morris, and T. Nguyen, Tree LANs with Collision Avoidance: Protocol and switch Architecture, in Proc. IEEE Globecom '87 (1987).

[4] T. Suda and S. Morris, Tree LANs with Collision Avoidance: Station and Switch Protocols, Computer Networks ISDN Systems 17 (2) (1989) 101-110.

[5] T. Suda, Thieu Q. Nguyen, and S. Morris, A Tree LAN with Collision Avoidance: Photonic Switch Design and Simulated Performance, Computer Networks ISDN Systems 17 (1) (1989) 89-100110.

[6] T. Suda, Y. Yemini and M. Schwartz, Tree Network with Collision Avoidance Switches, in: Proc. IEEE Infocom '84 (1984).

[7] T. Suda, H. Hung, and K. Godbille, TTL Implementations of a CAMB Tree Switch, Project Report.

[8] Silicon Complier Systems Corporation, 1988, 1989.

10 Appendix

This appendix contains schematic diagrams, layout, and test waveforms for a CAMB switch implementation.

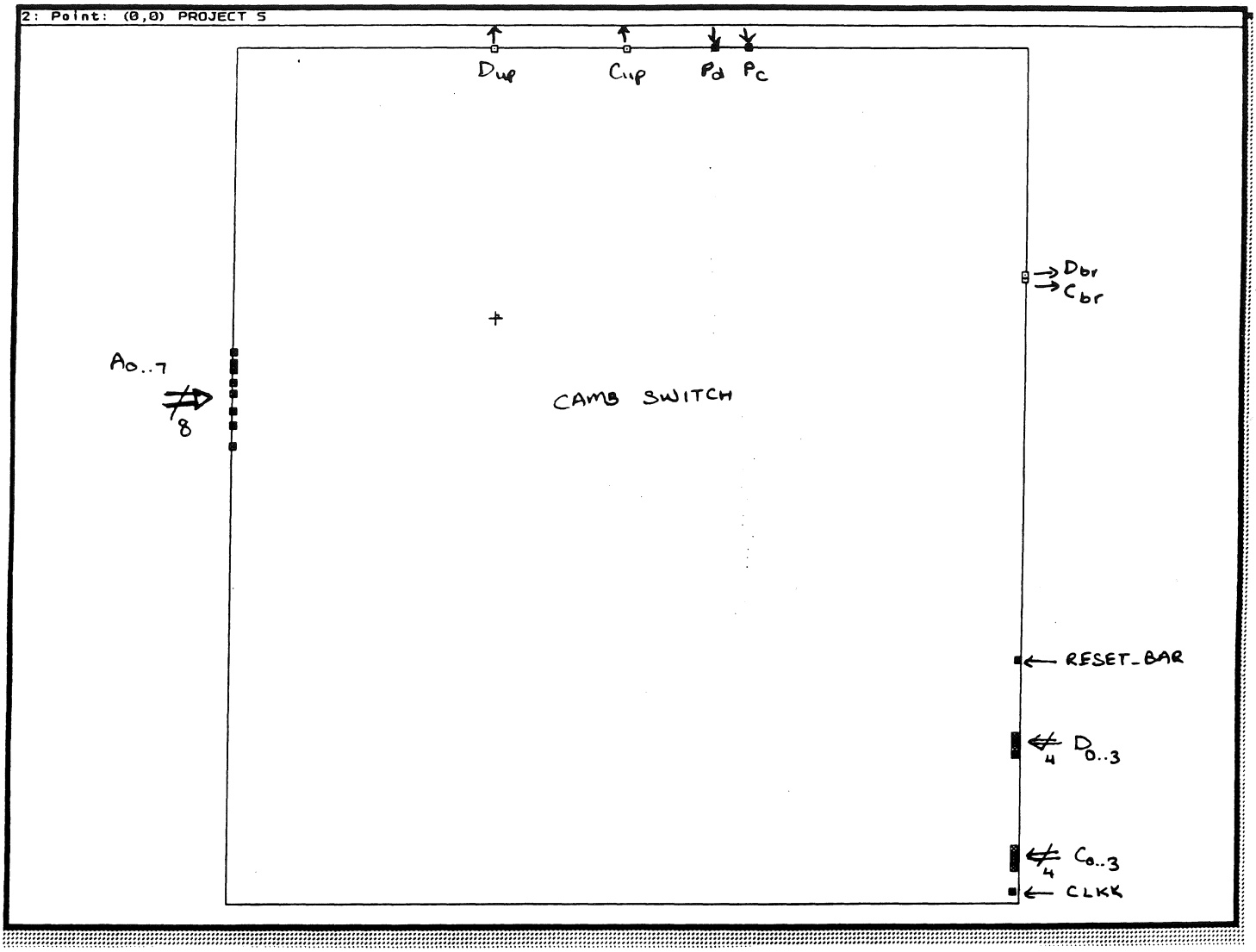

Figure 15: External I/O Interface of CAMB Switch

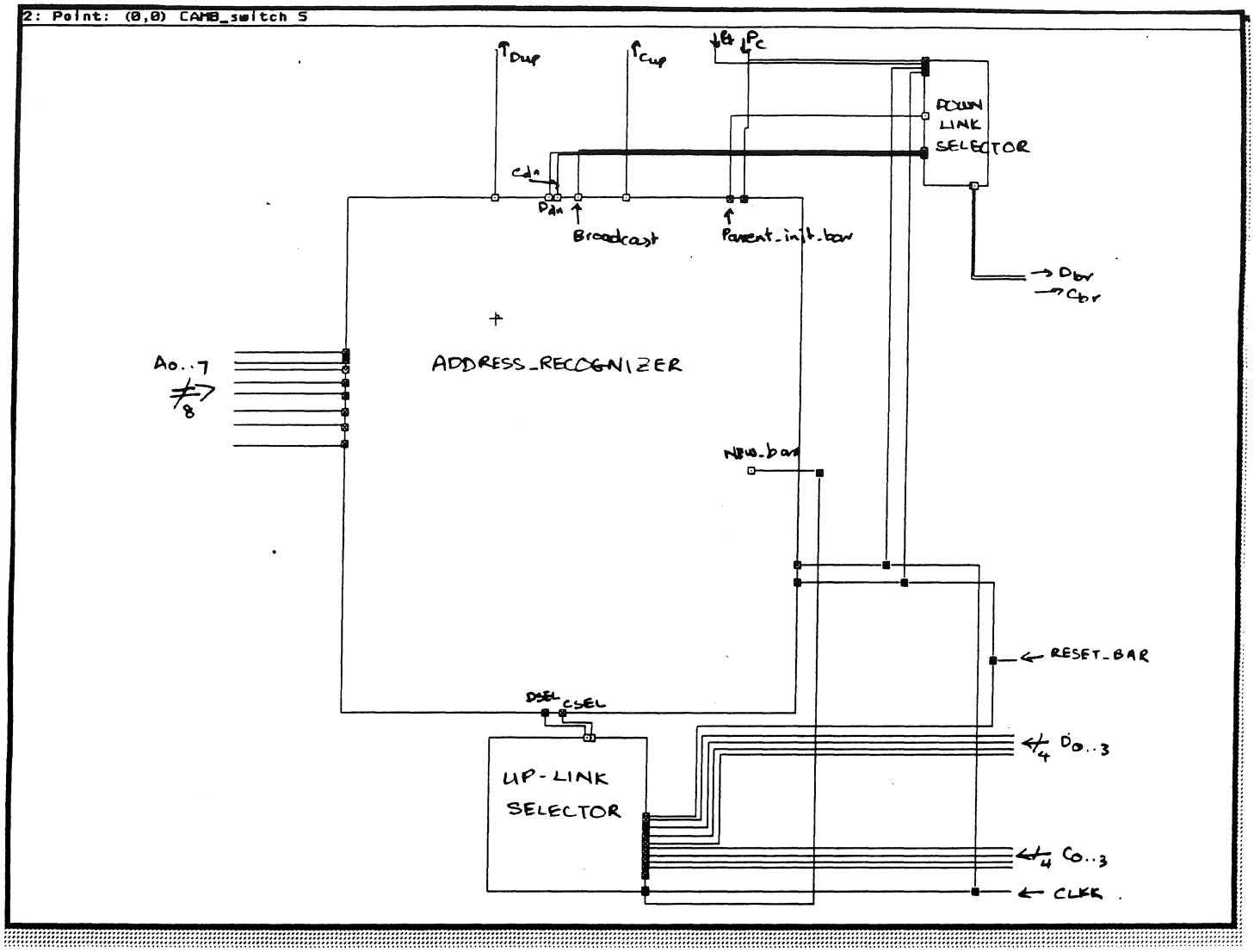

Figure 16: CAMB Switch Schematic Diagram

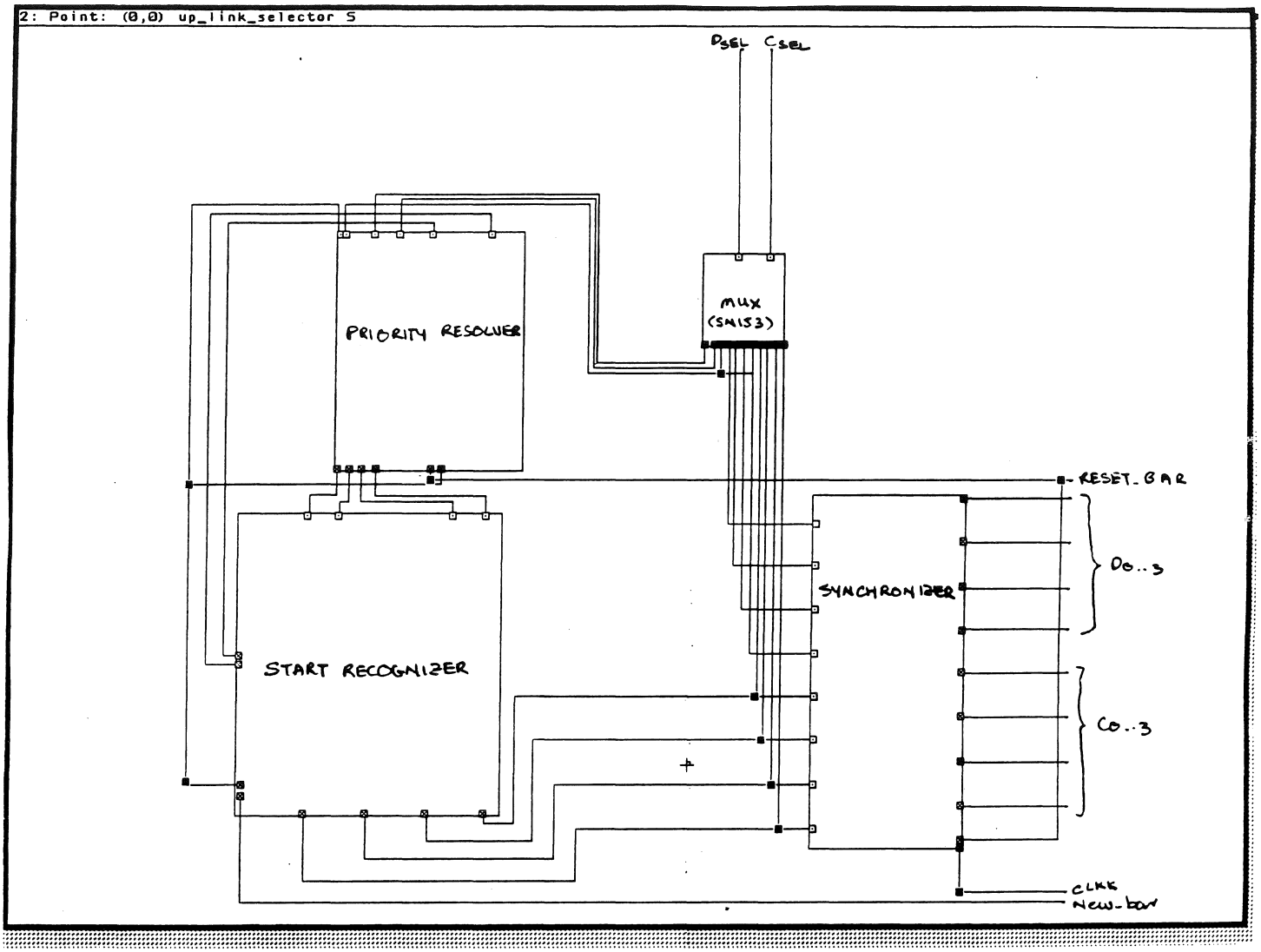

Figure 17: Uplink Selector Schematic Diagram

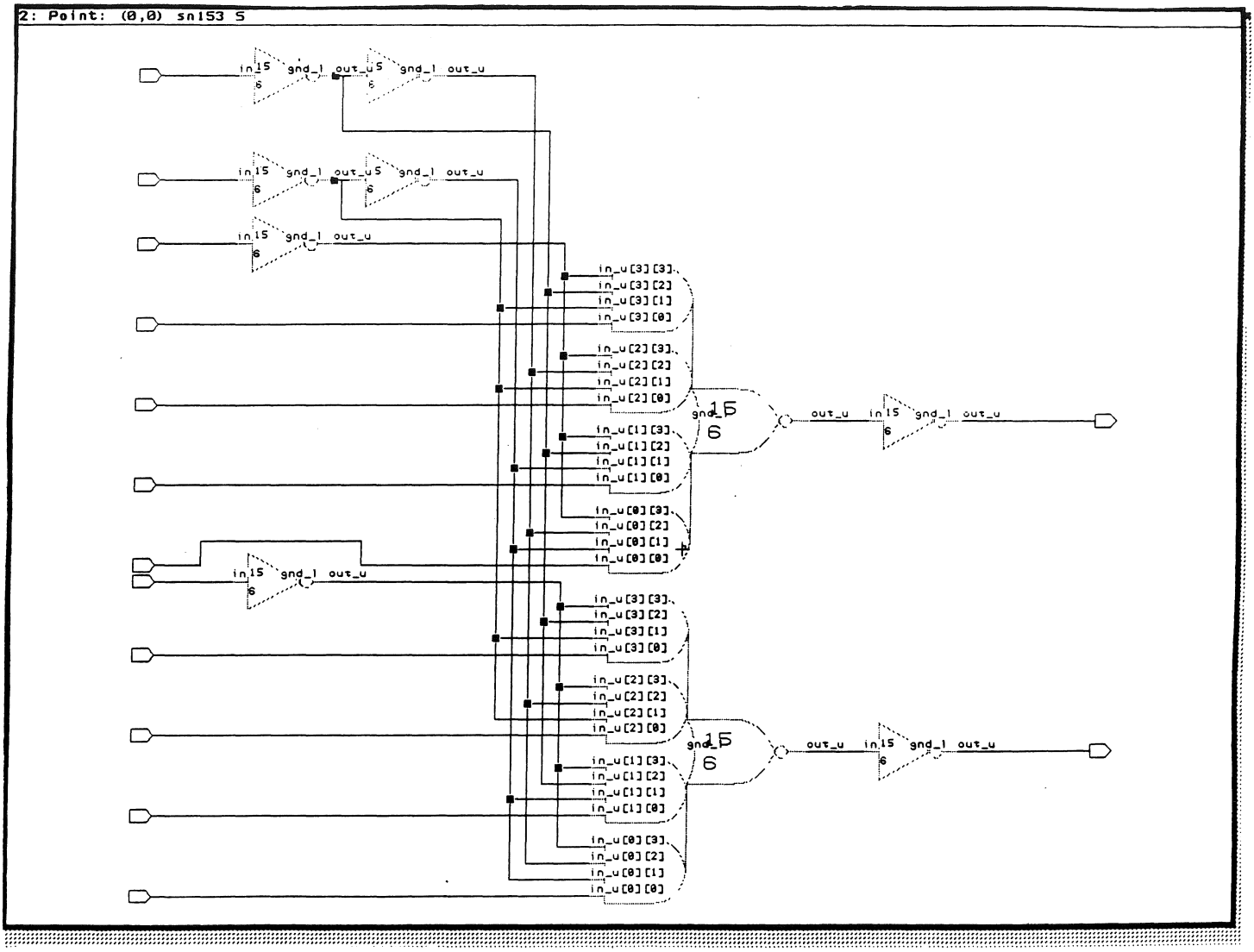

Figure 18: Mux (SN153) Schematic Diagram

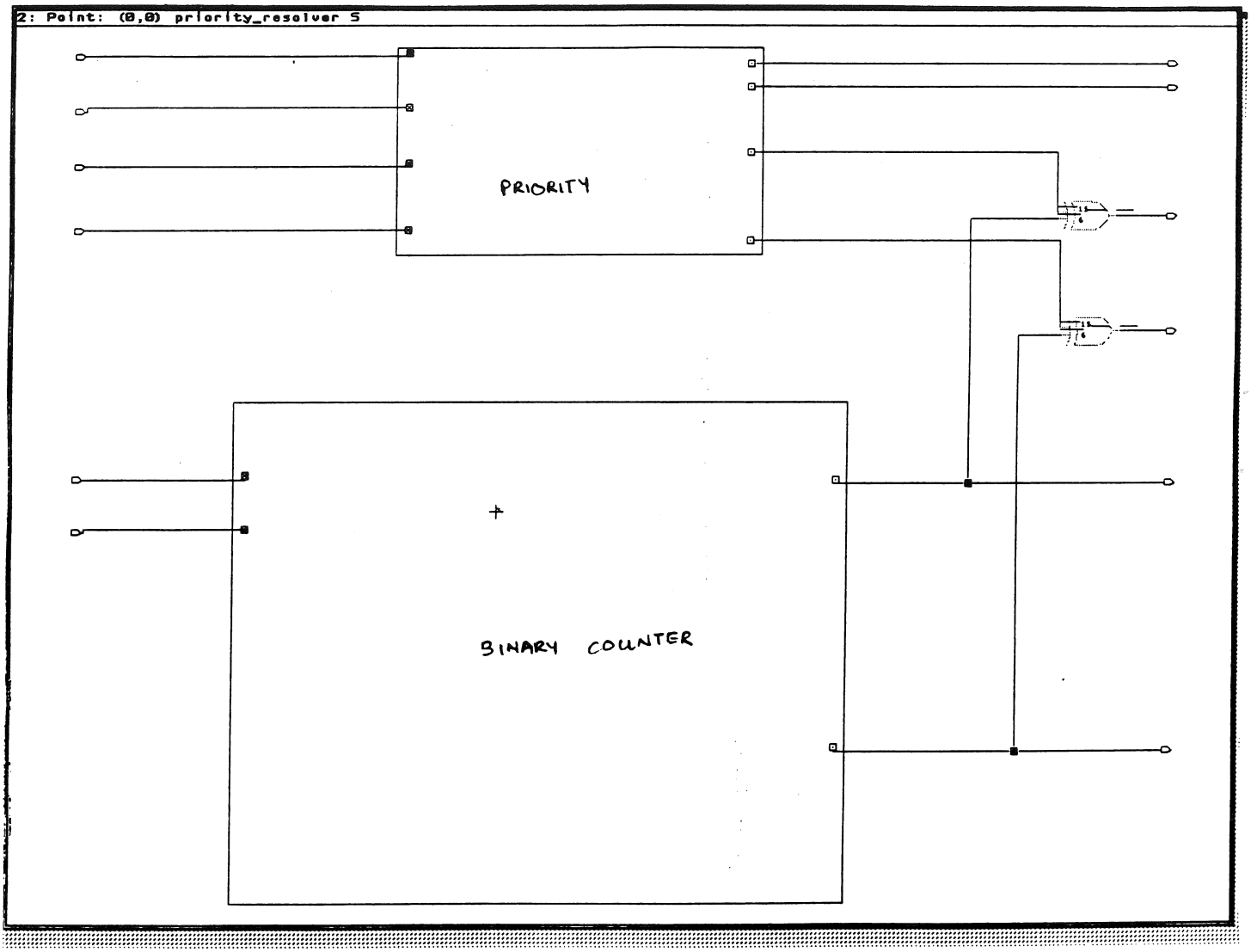

Figure 19: Priority Resolver Schematic Diagram

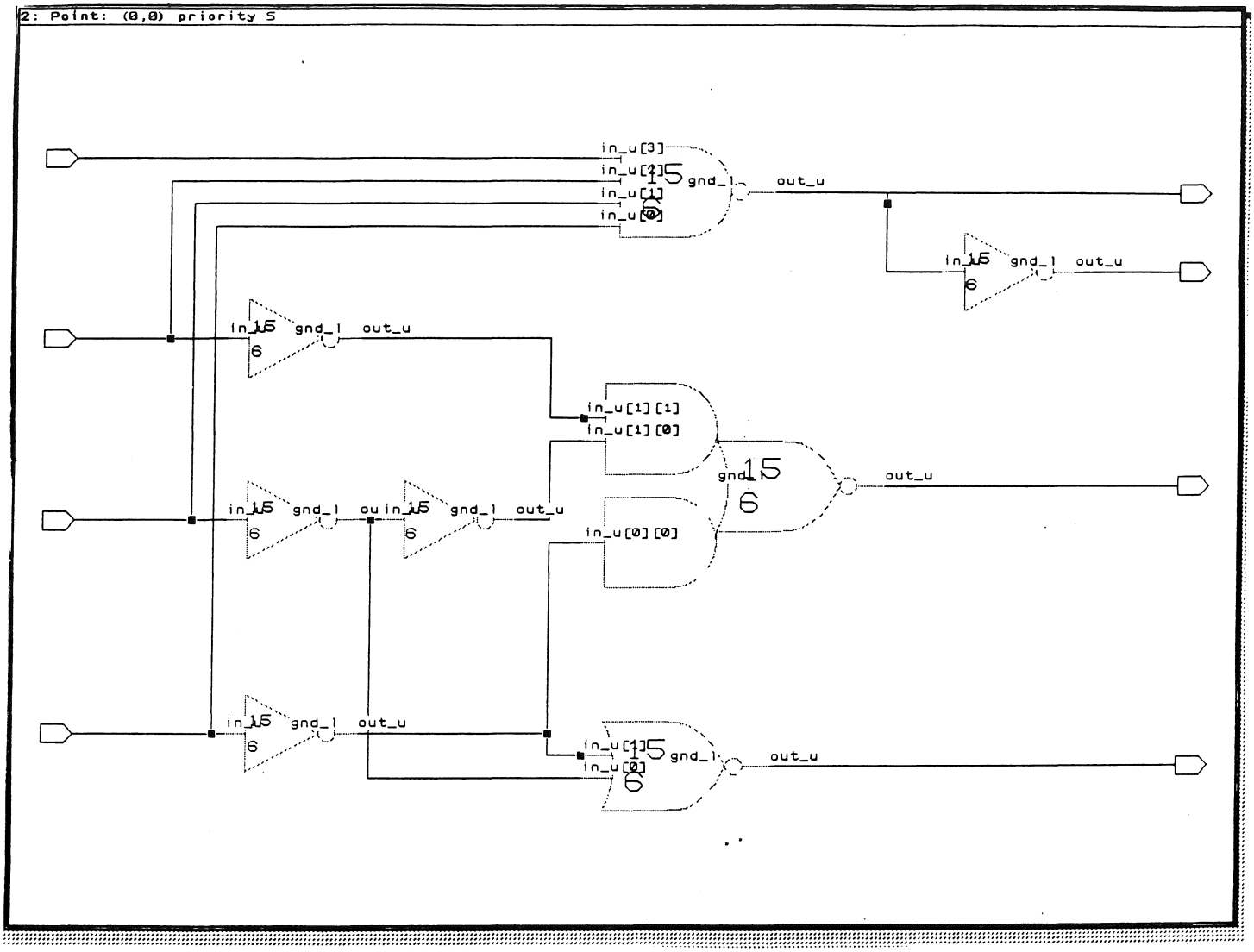

Figure 20: Priority Schematic Diagram

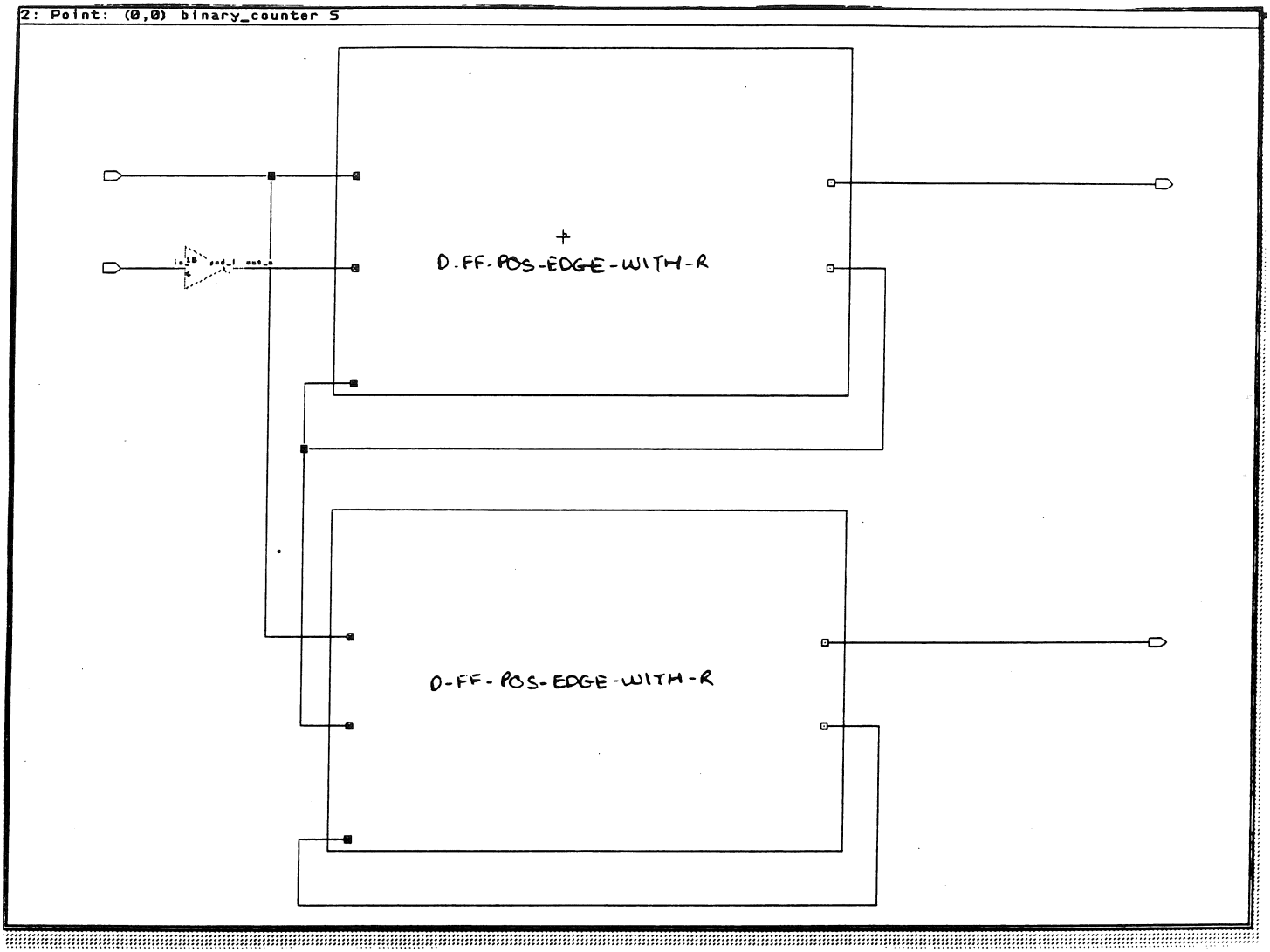

Figure 21: Binary Counter Schematic Diagram

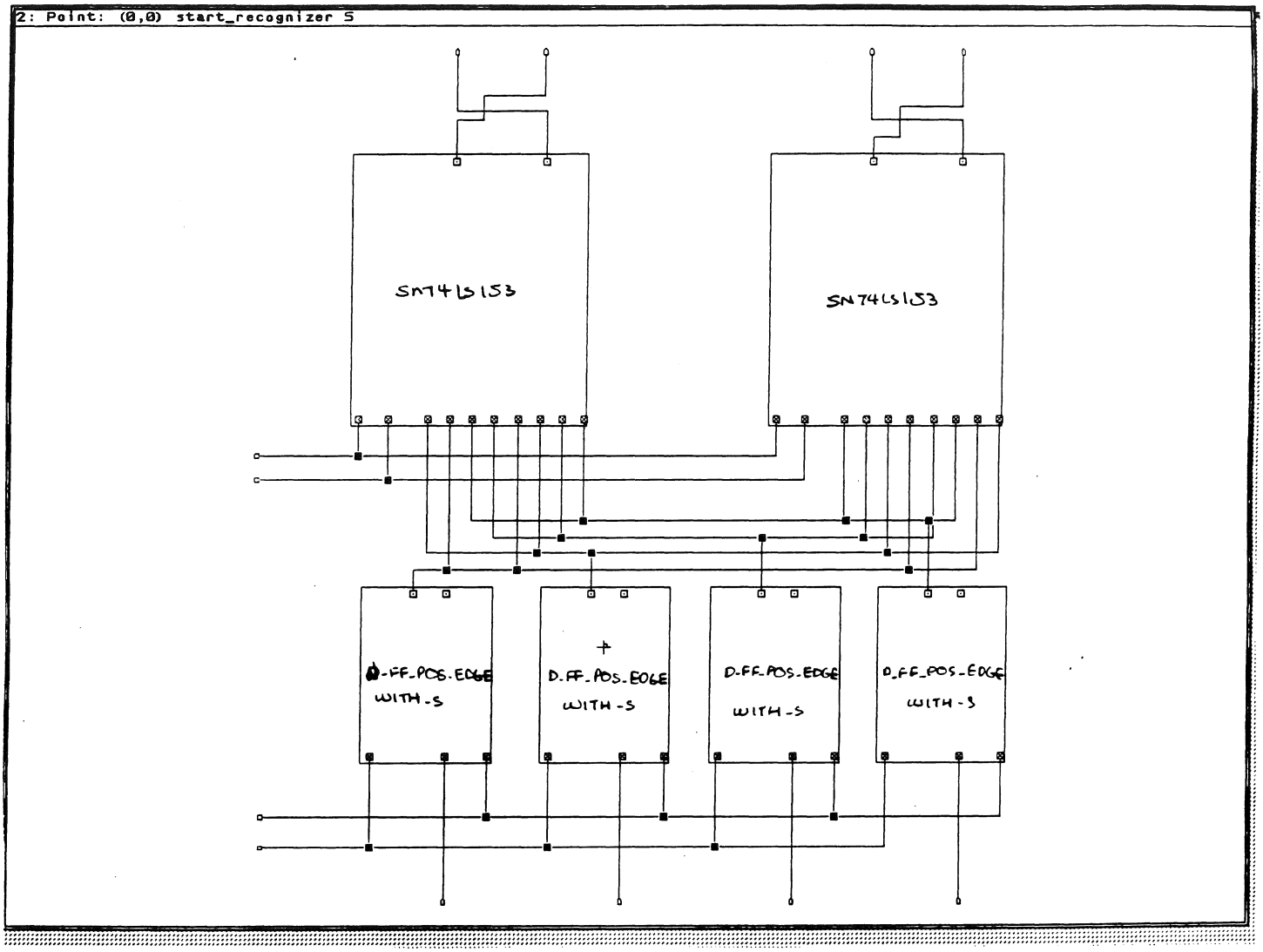

Figure 22: Start Recognizer Schematic Diagram

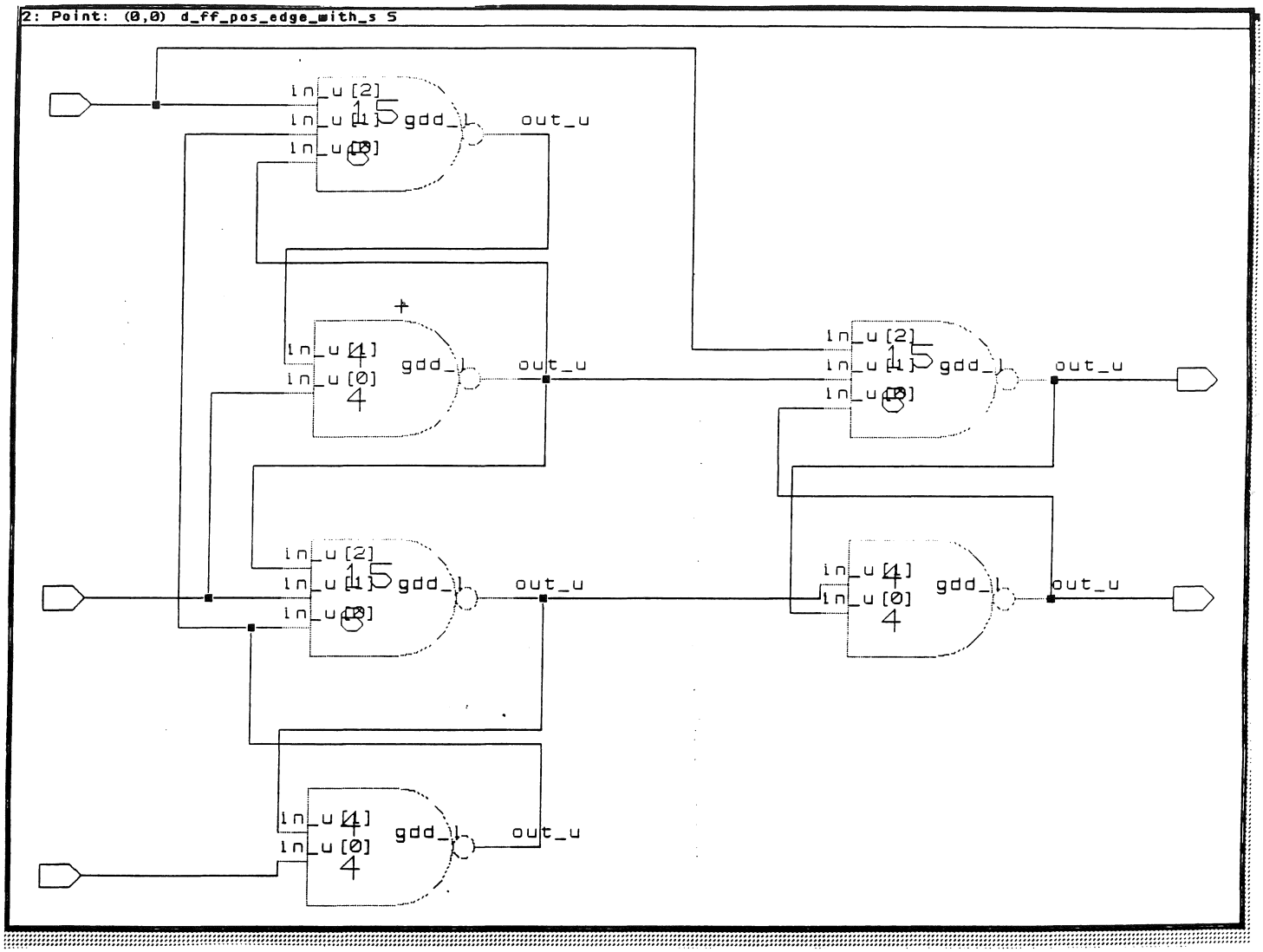

Figure 23: DFF Pos-Edge With Set Schematic Diagram

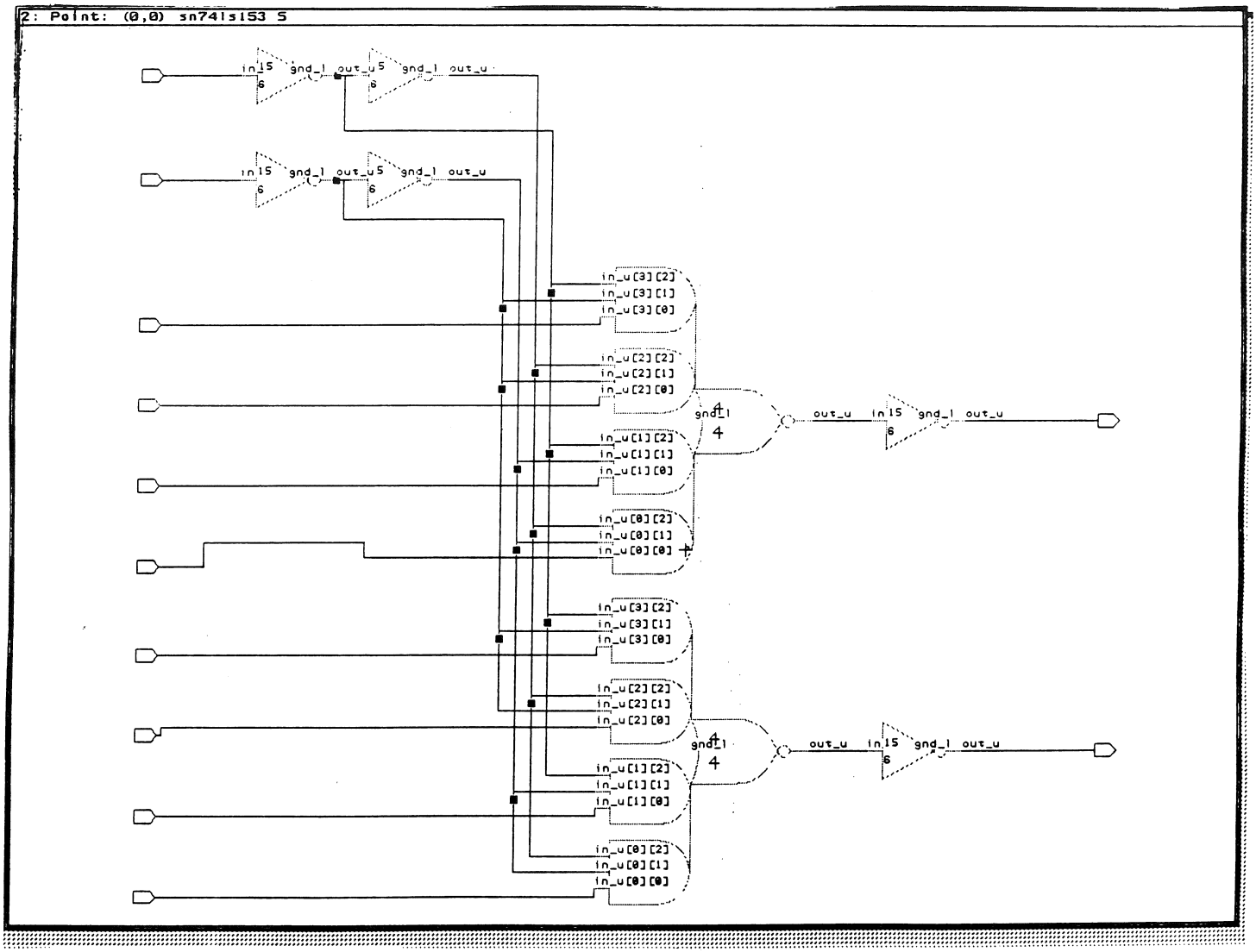

Figure 24: SN74LS153 Schematic Diagram

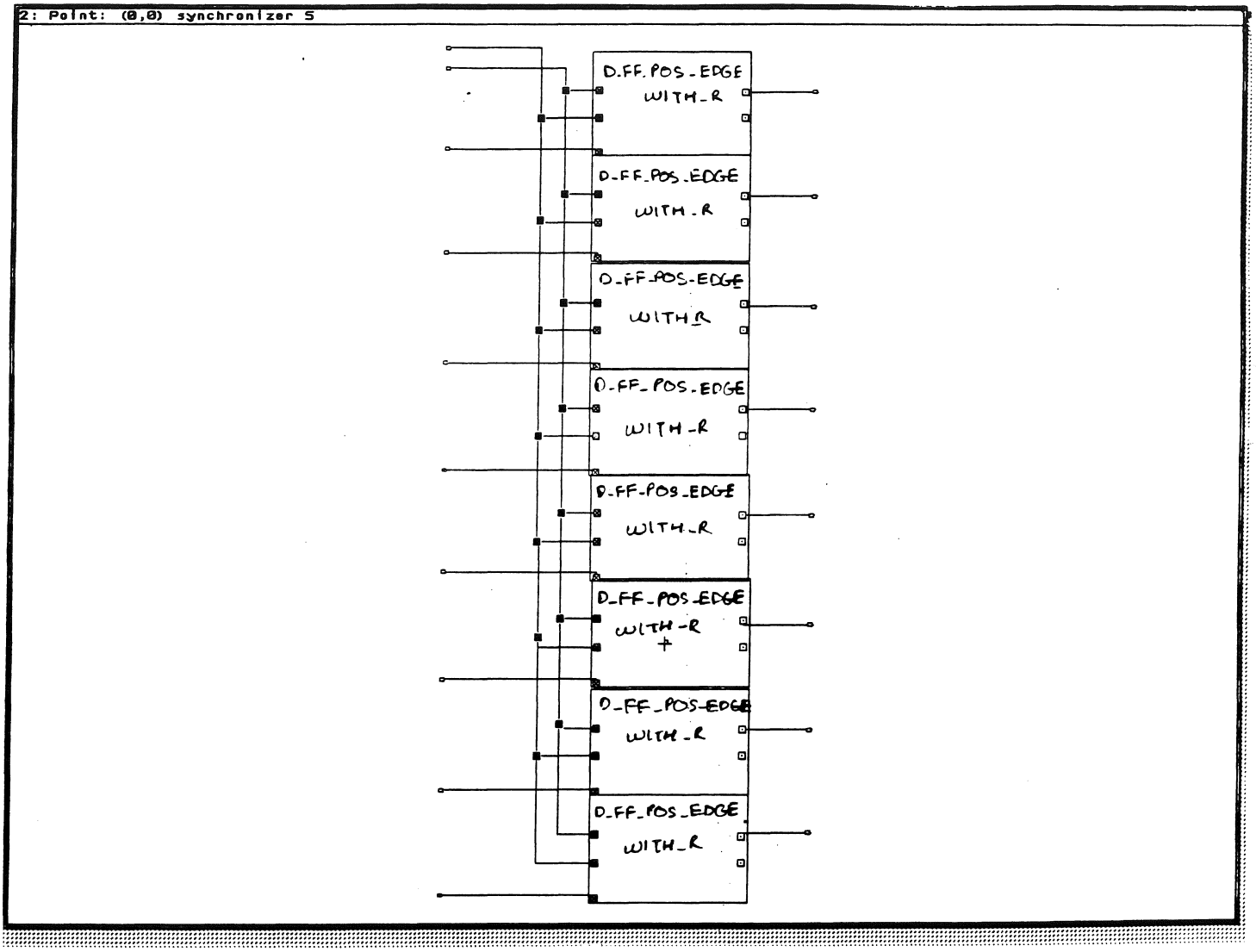

Figure 25: Synchronizer Schematic Diagram

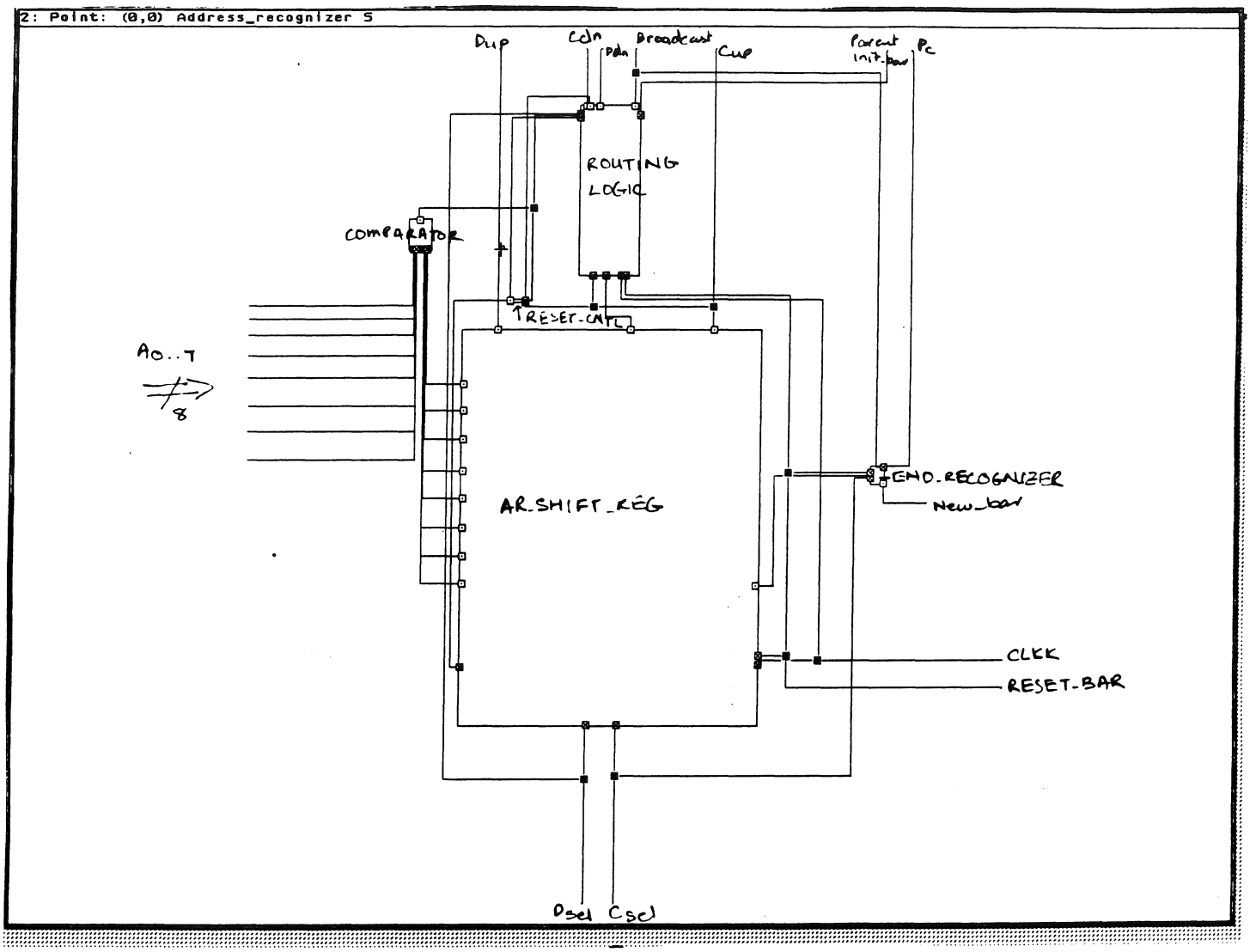

Figure 26: Address Recognizer Schematic Diagram

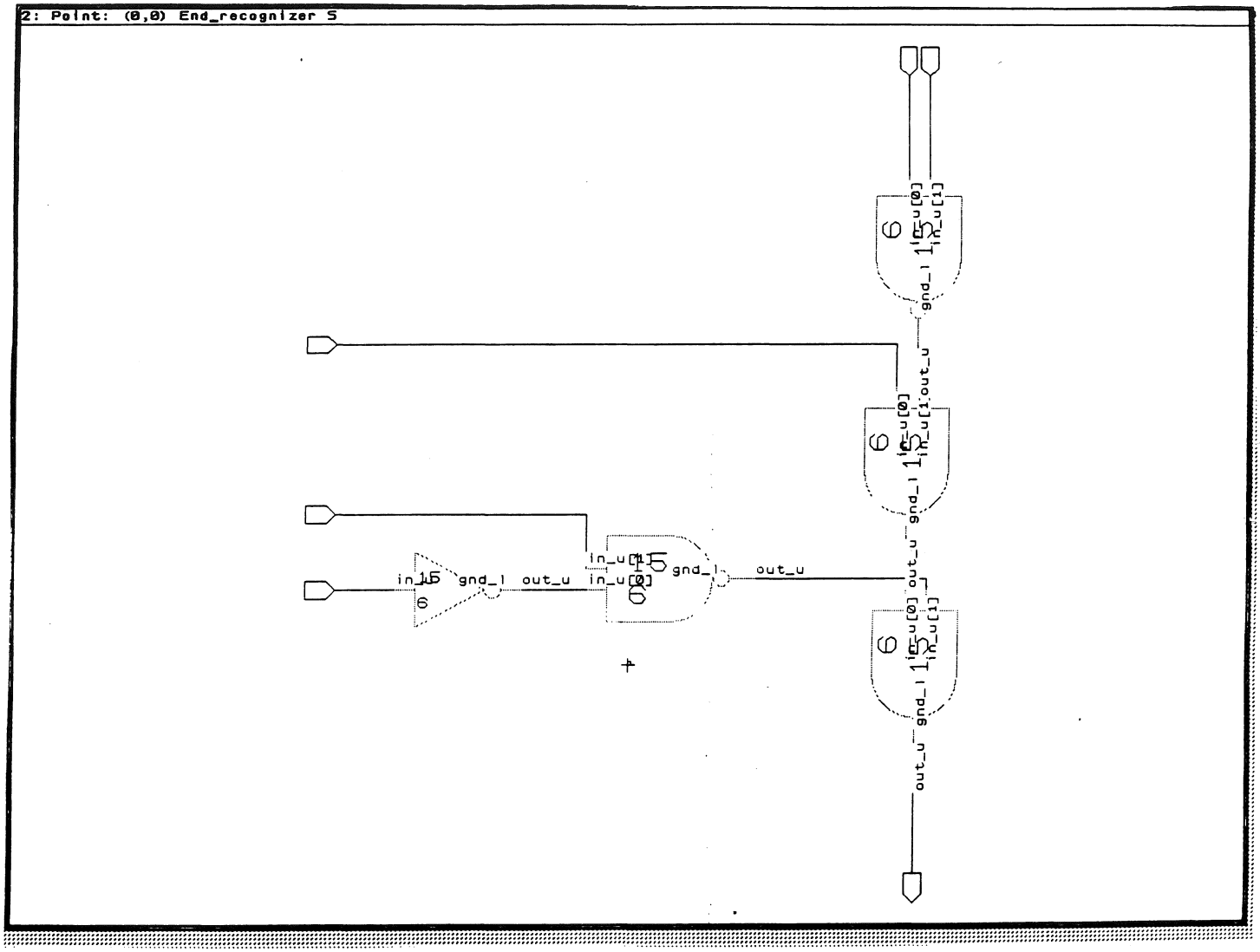

Figure 27: End Recognizer Schematic Diagram

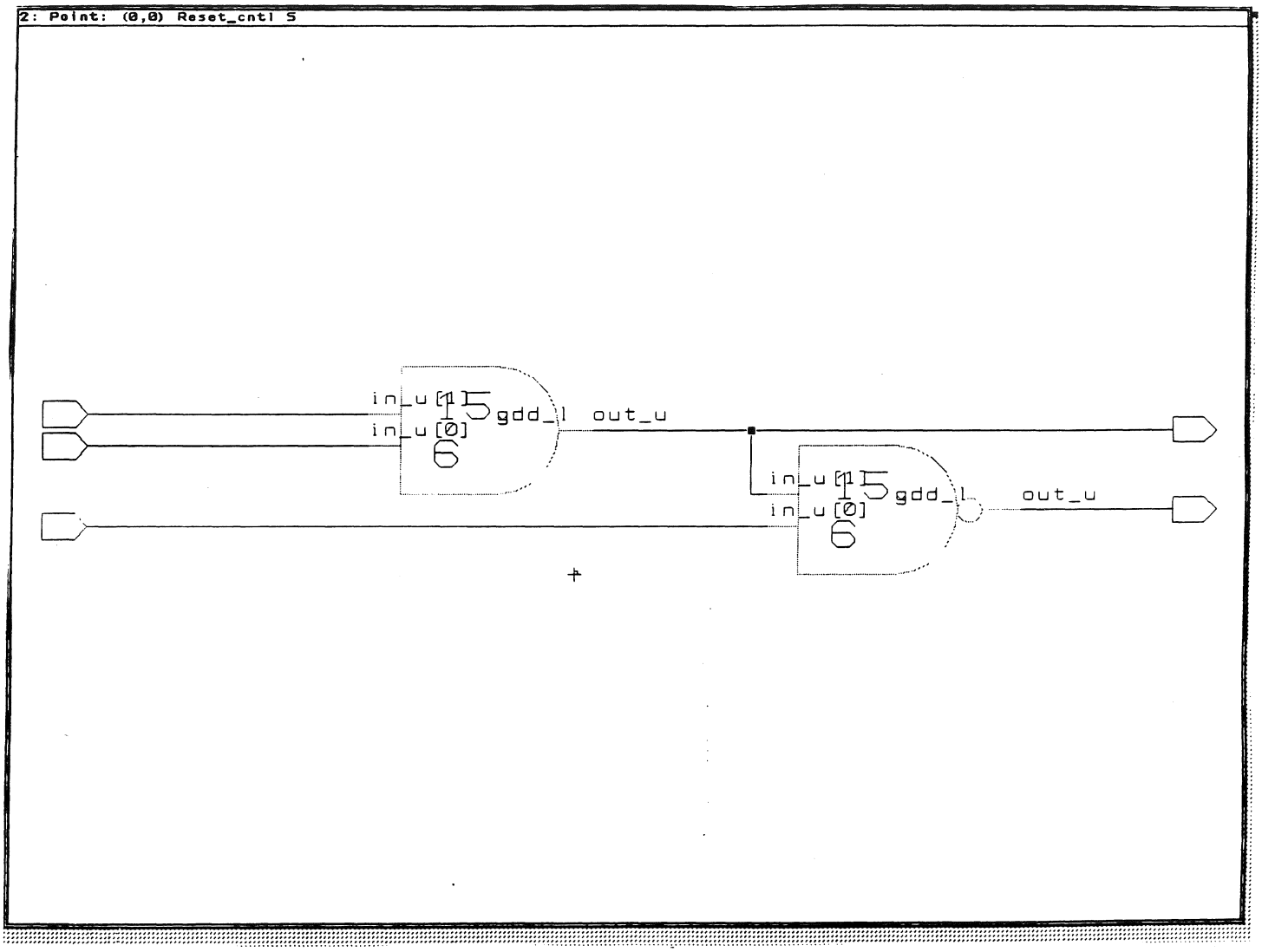

Figure 28: Reset Controller Schematic Diagram

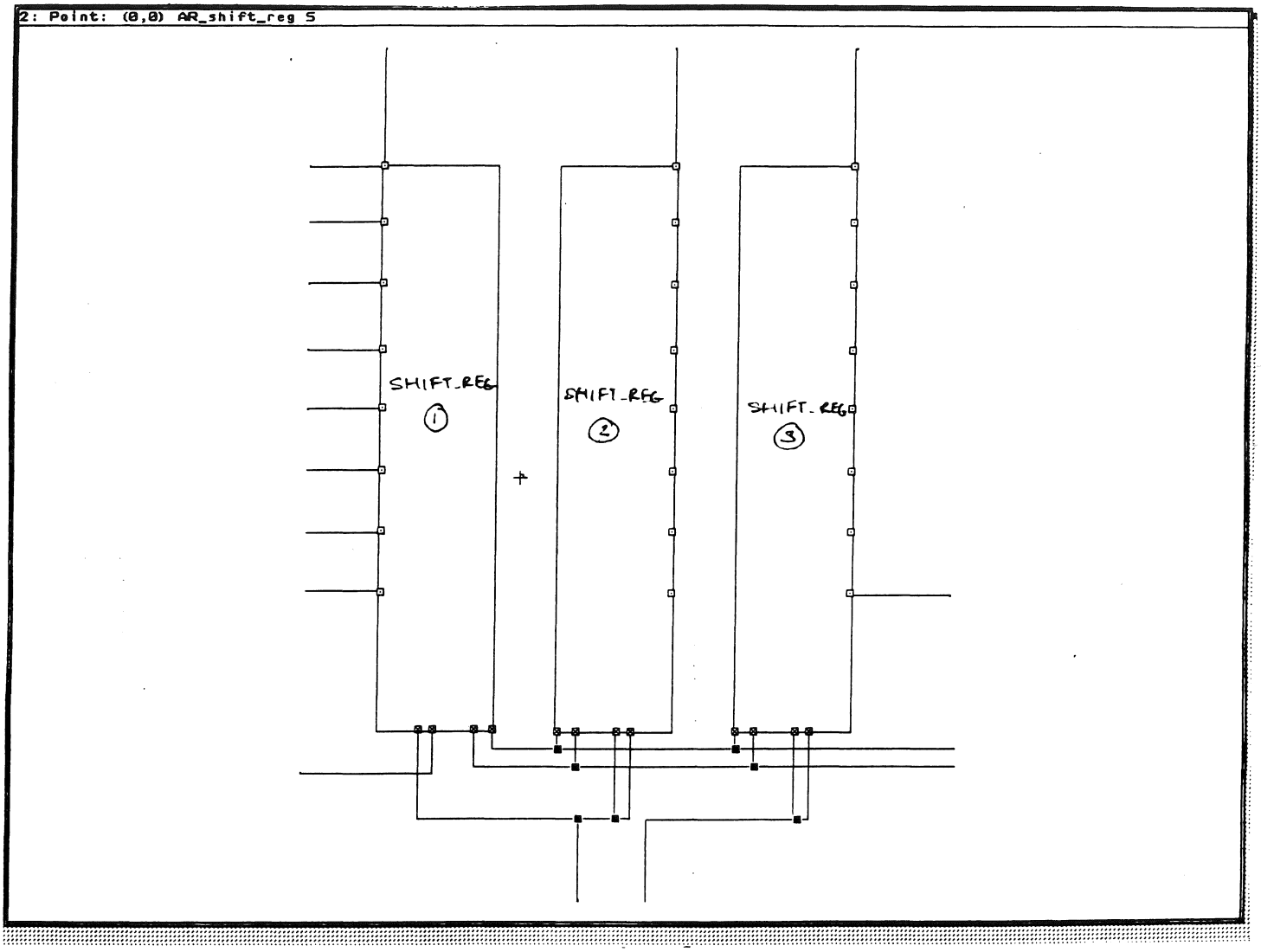

Figure 29: AR-shift Register Schematic Diagram

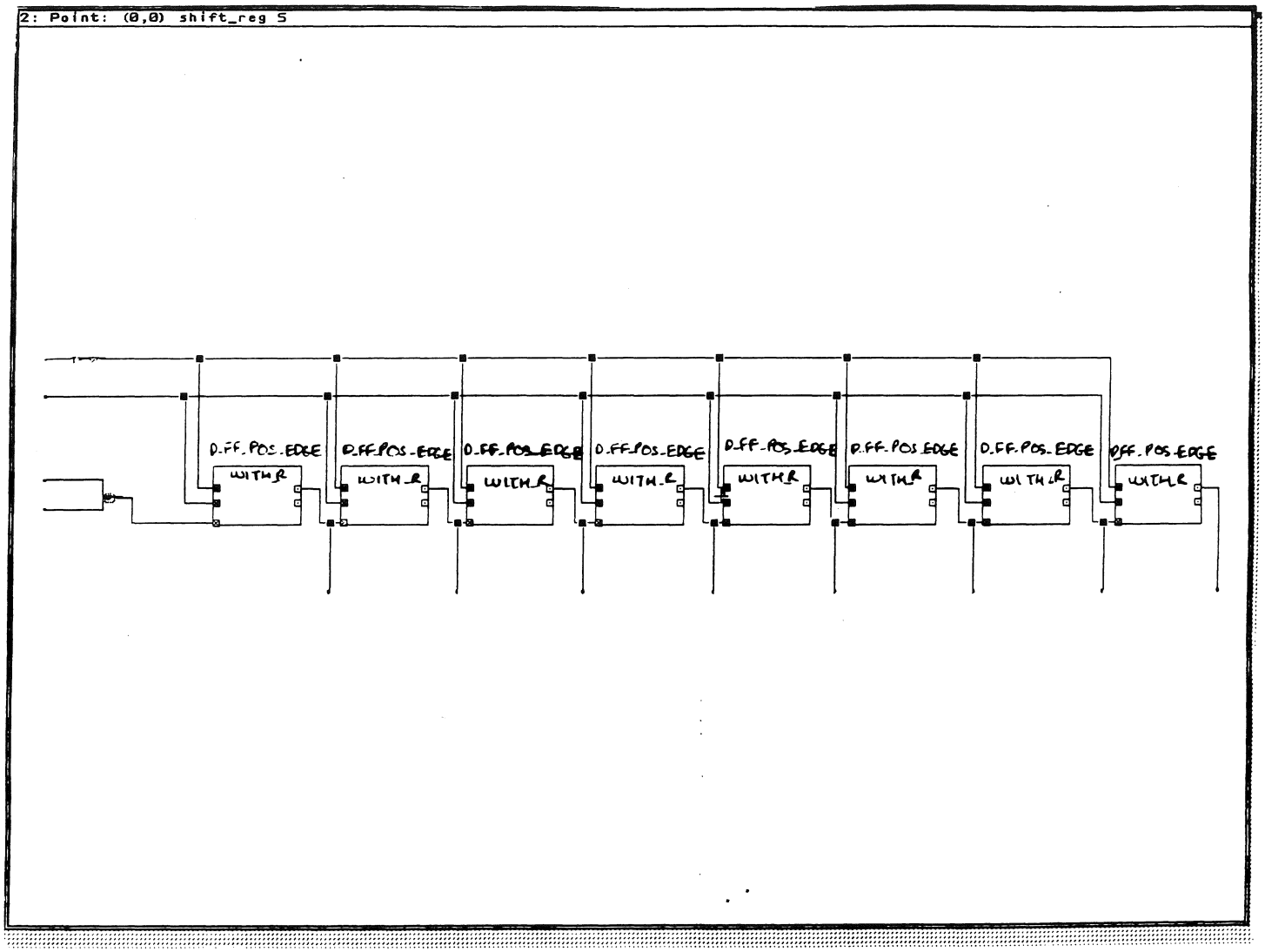

Figure 30: Shift Register Schematic Diagram

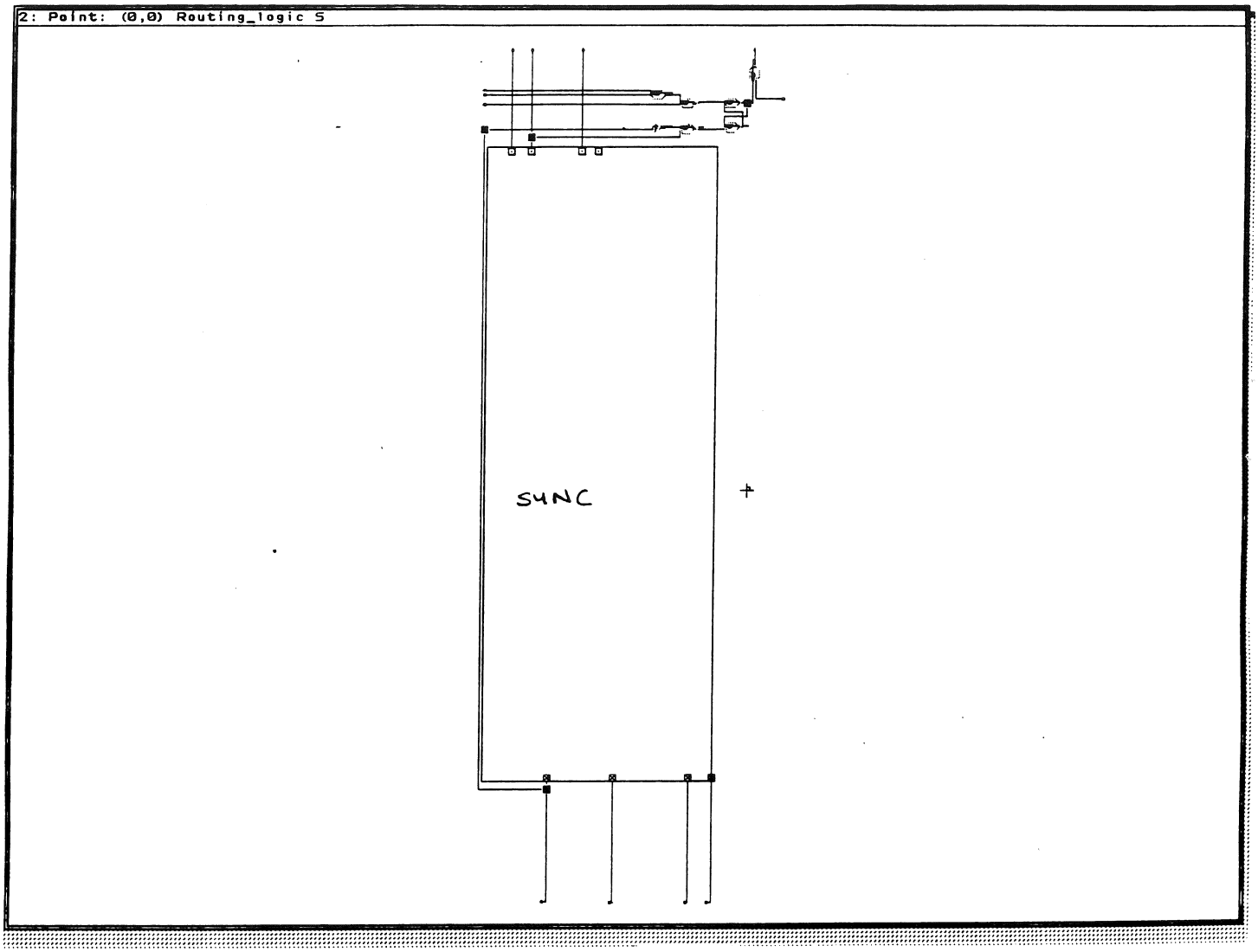

Figure 31: Routing Logic Schematic Diagram

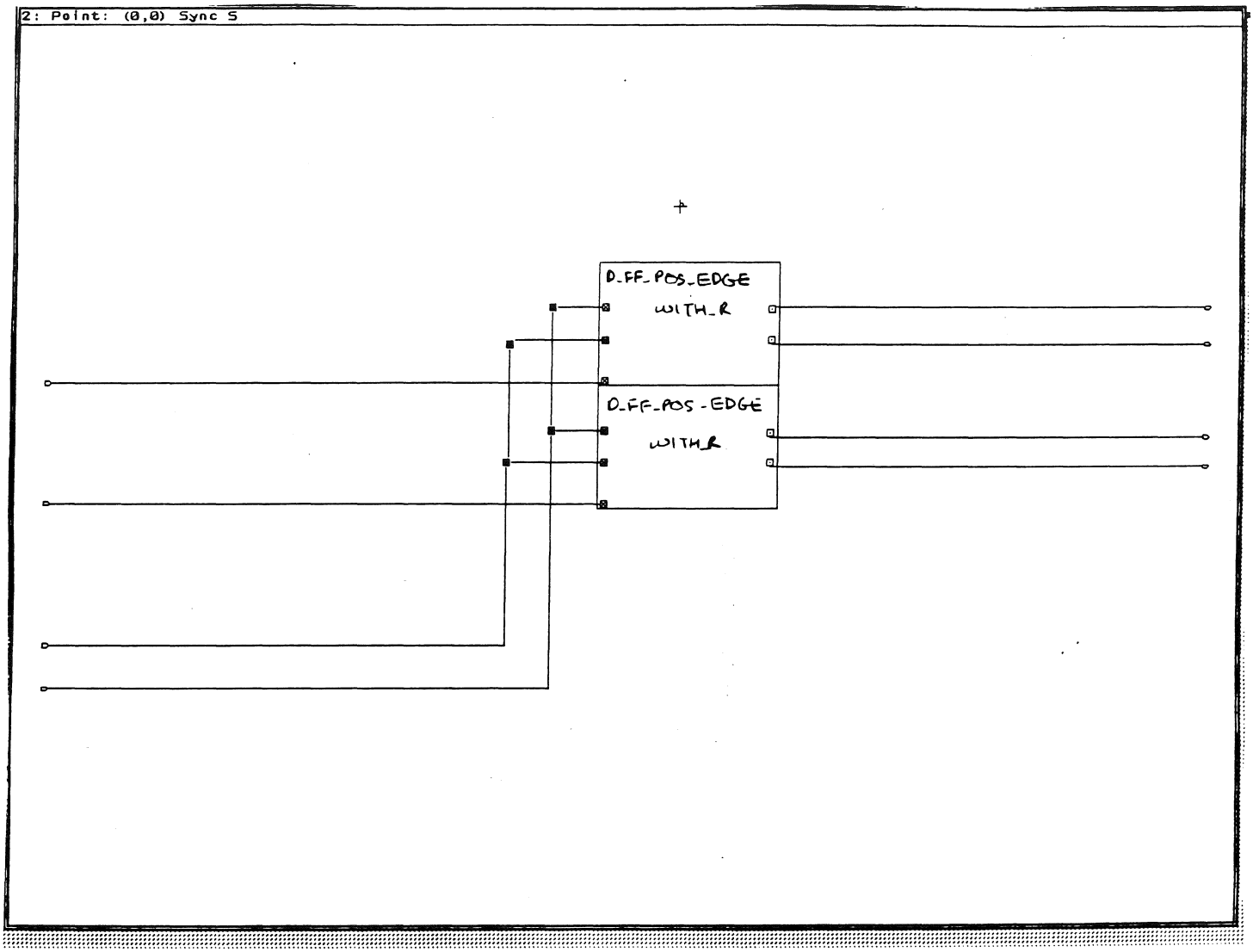

Figure 32: Sync Schematic Diagram

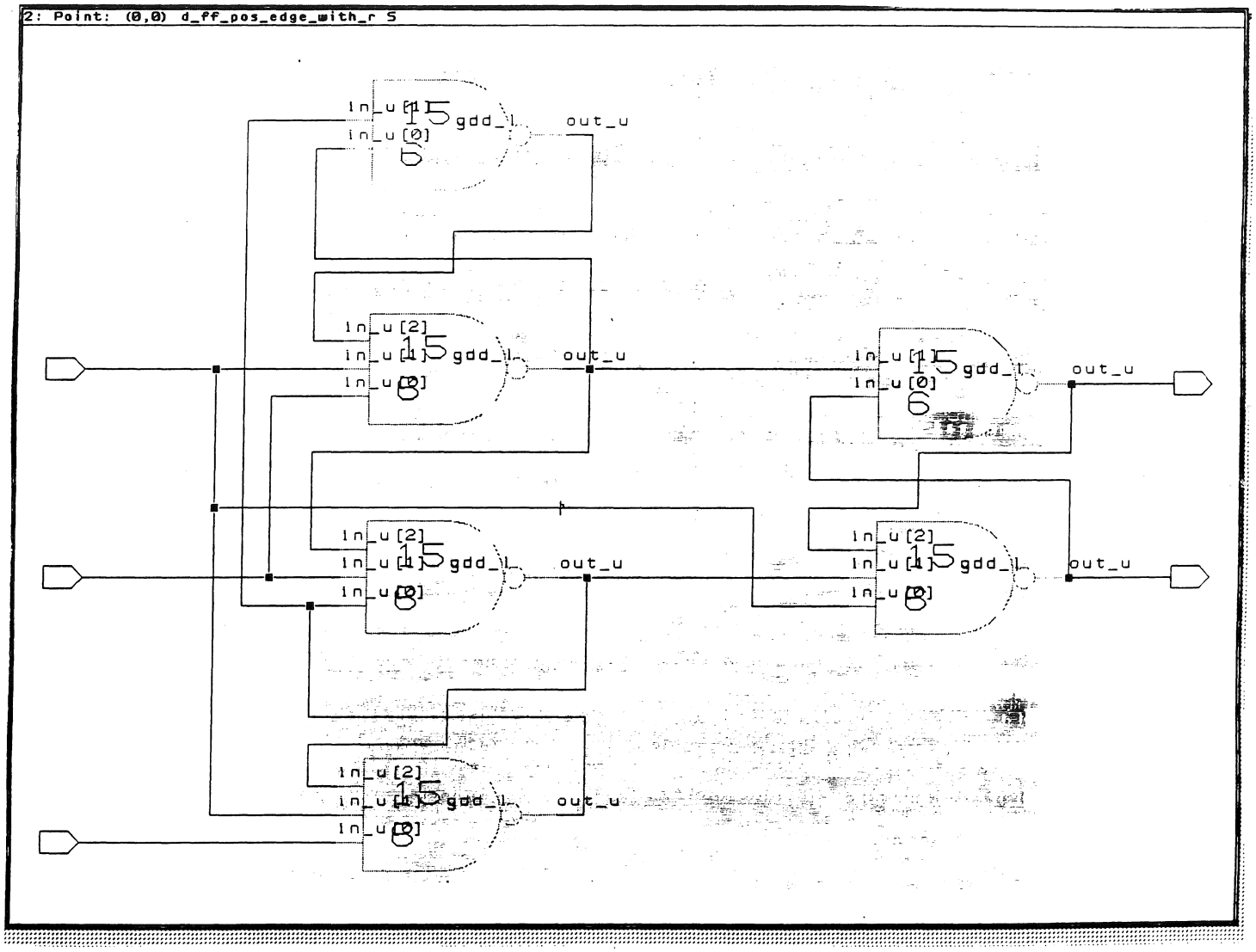

Figure 33: DFF Pos-Edge-Triggered With Reset Schematic Diagram

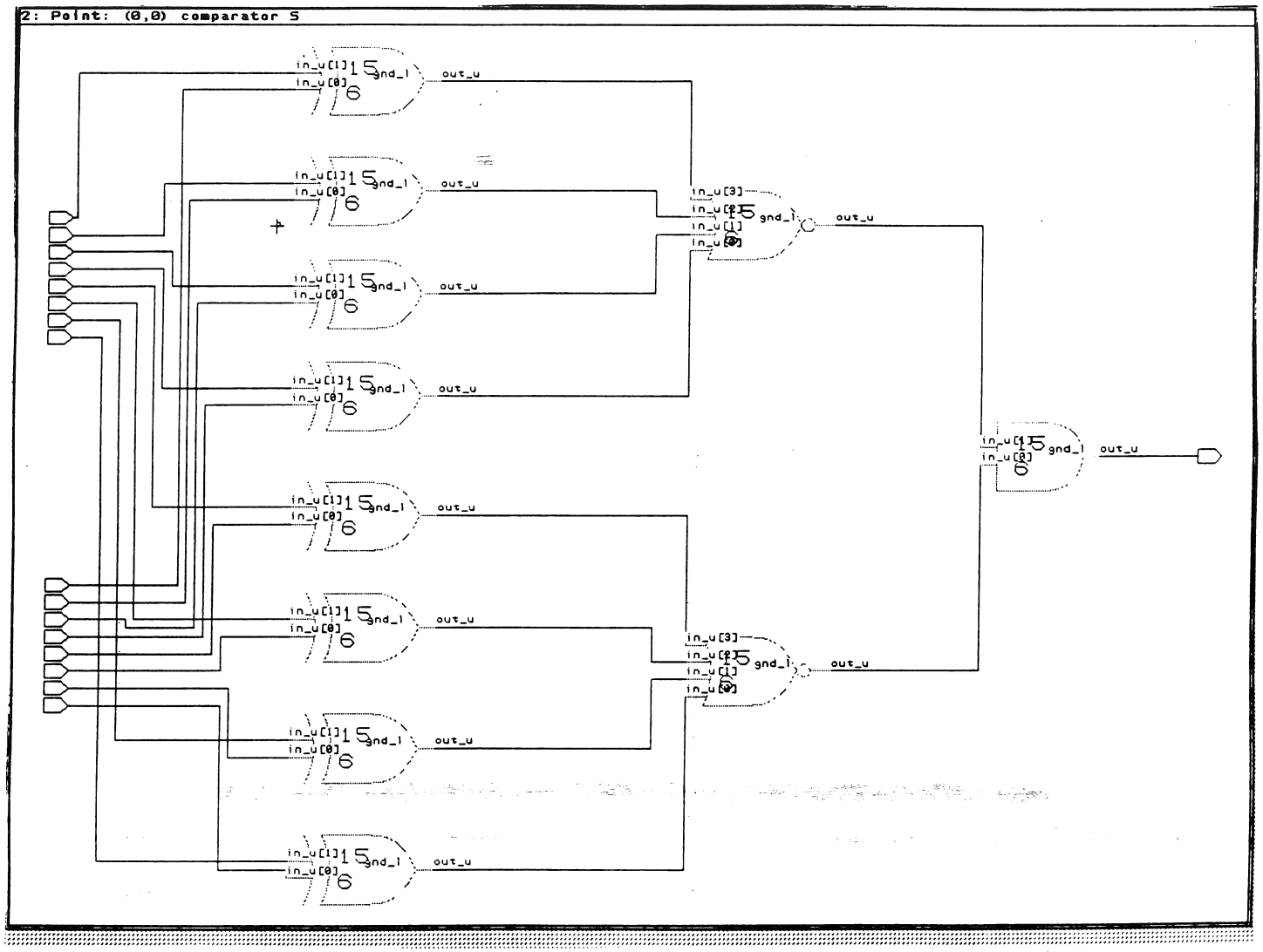

Figure 34: Comparator Schematic Diagram

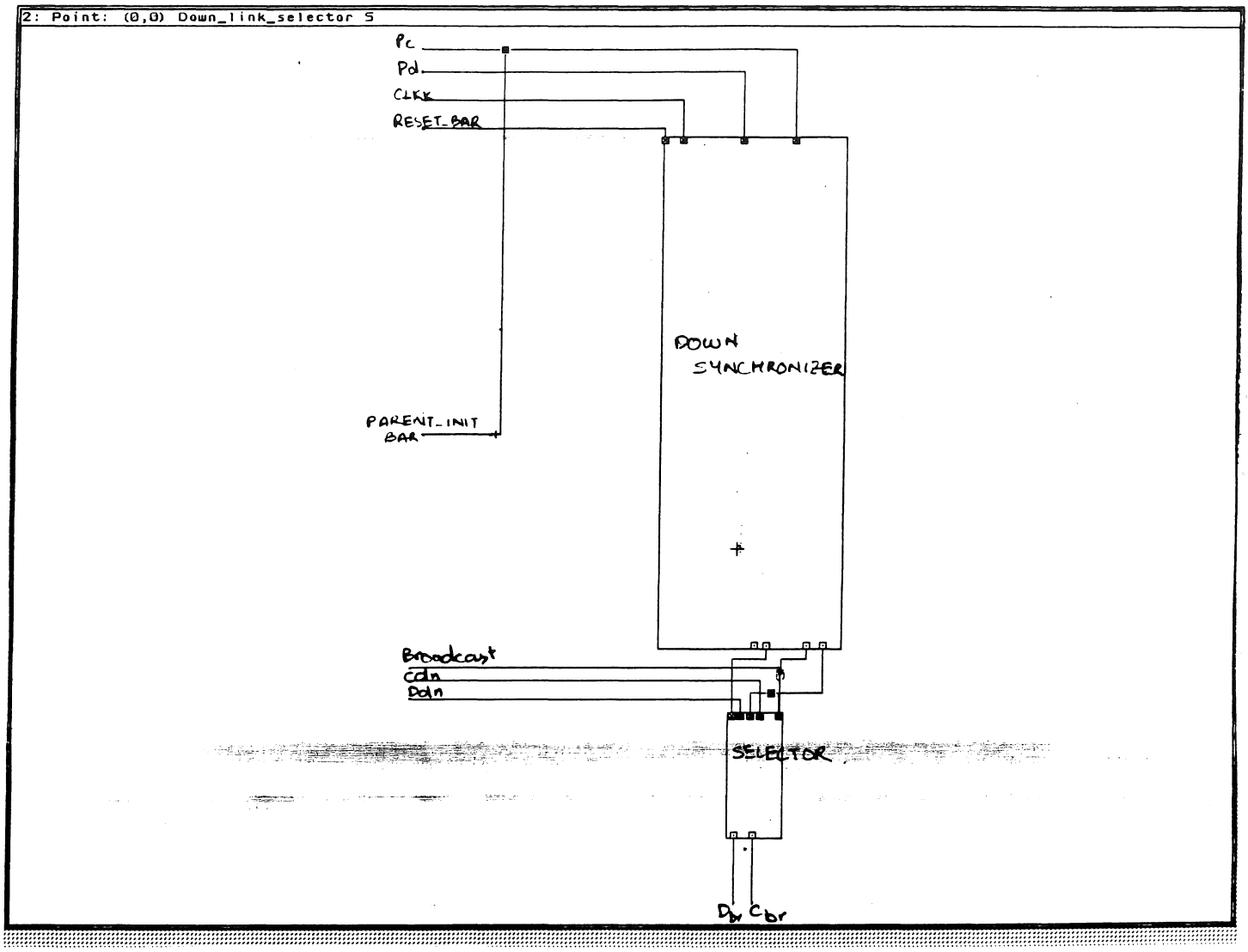

Figure 35: Downlink Selector Schematic Diagram

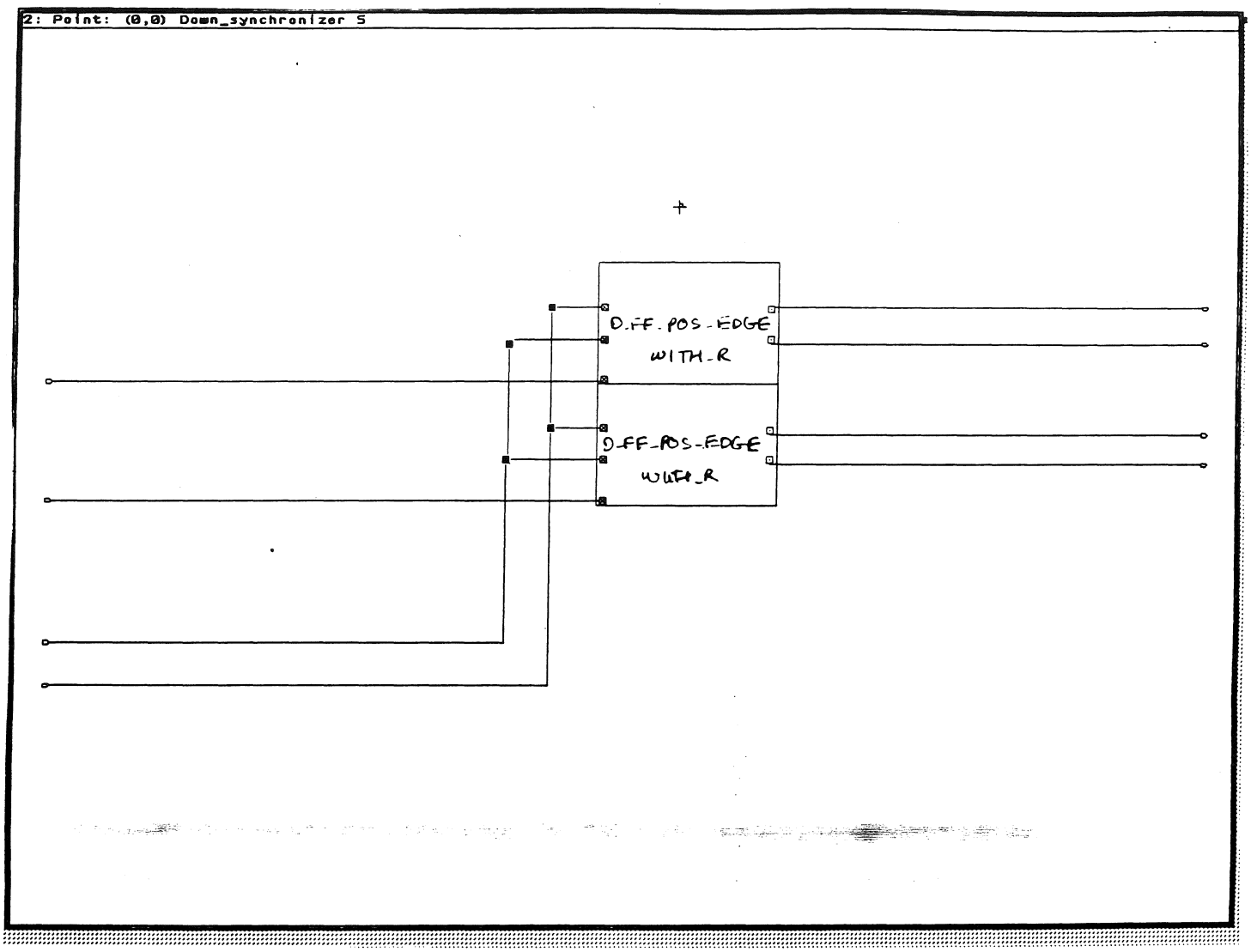

Figure 36: Down-Synchronizer Schematic Diagram

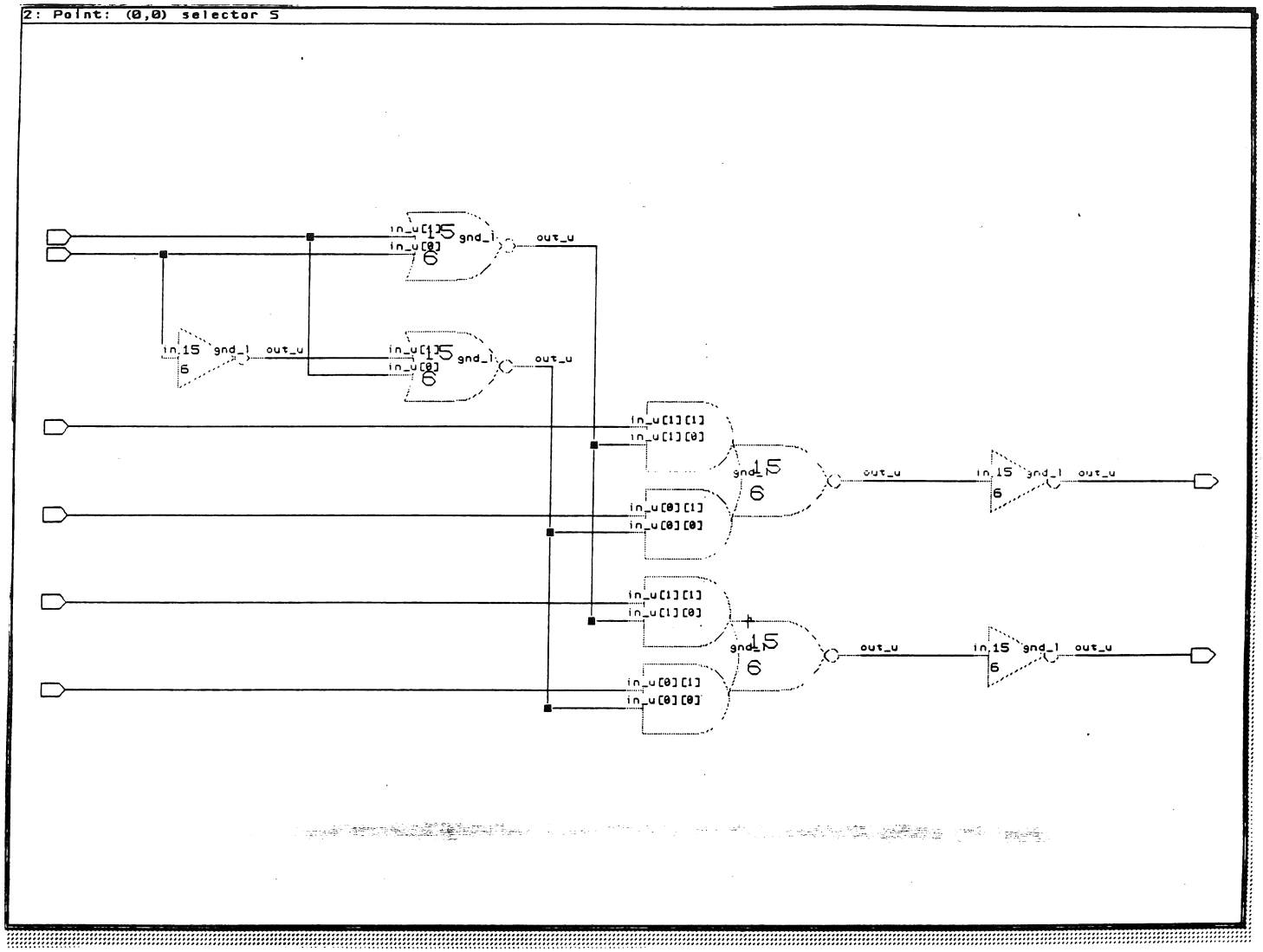

Figure 37: Selector Schematic Diagram

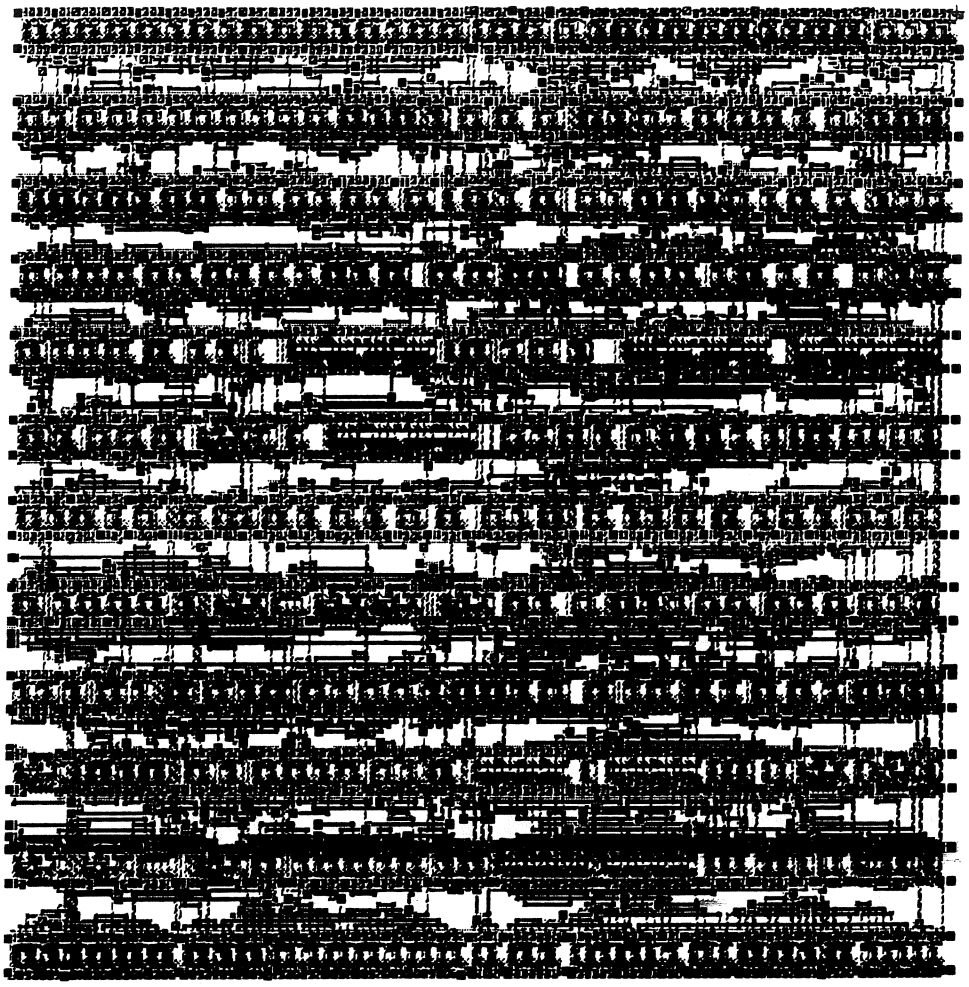

Figure 38: Layout of the CAMB Switch

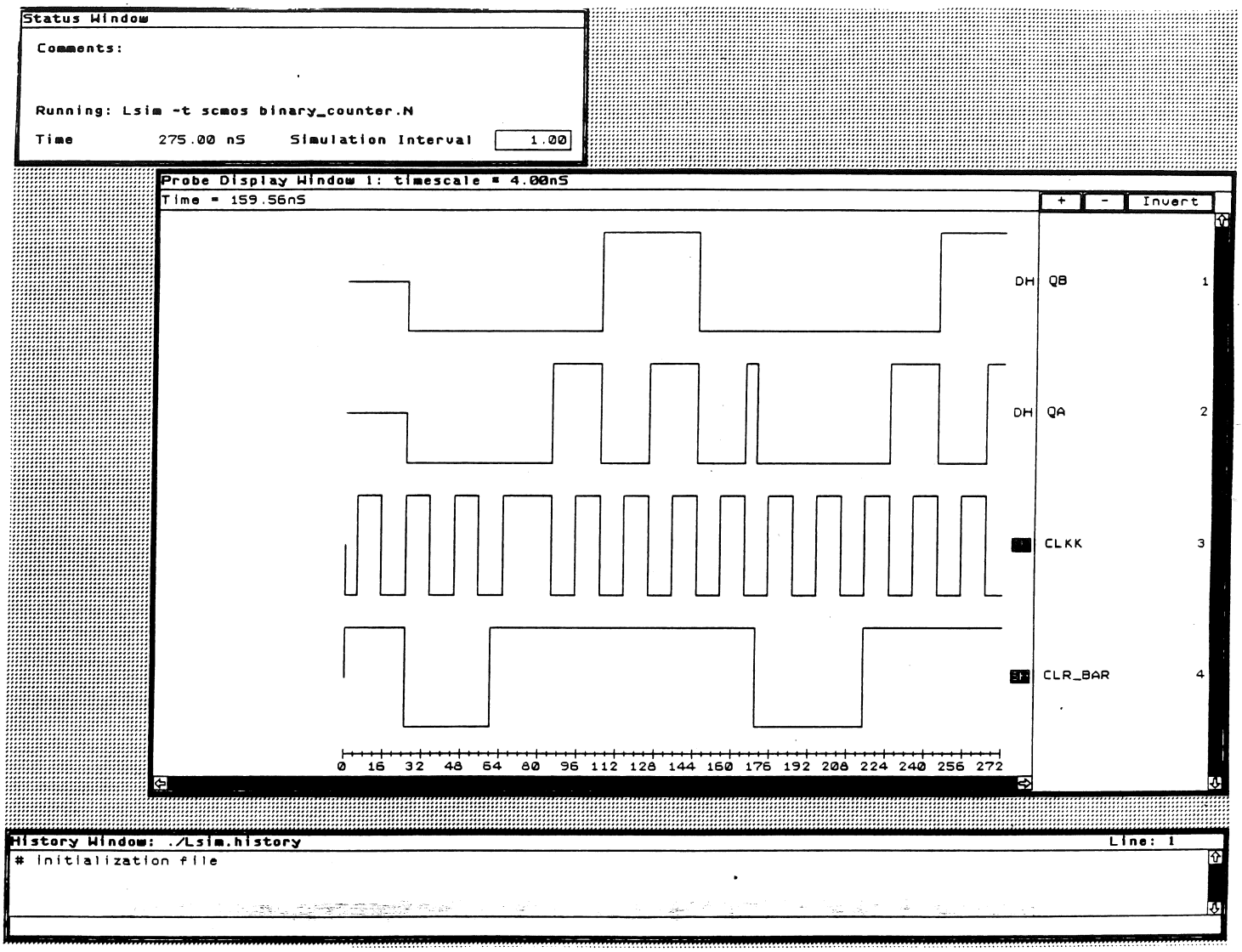

Figure 39: Test Waveform For Binary Counter

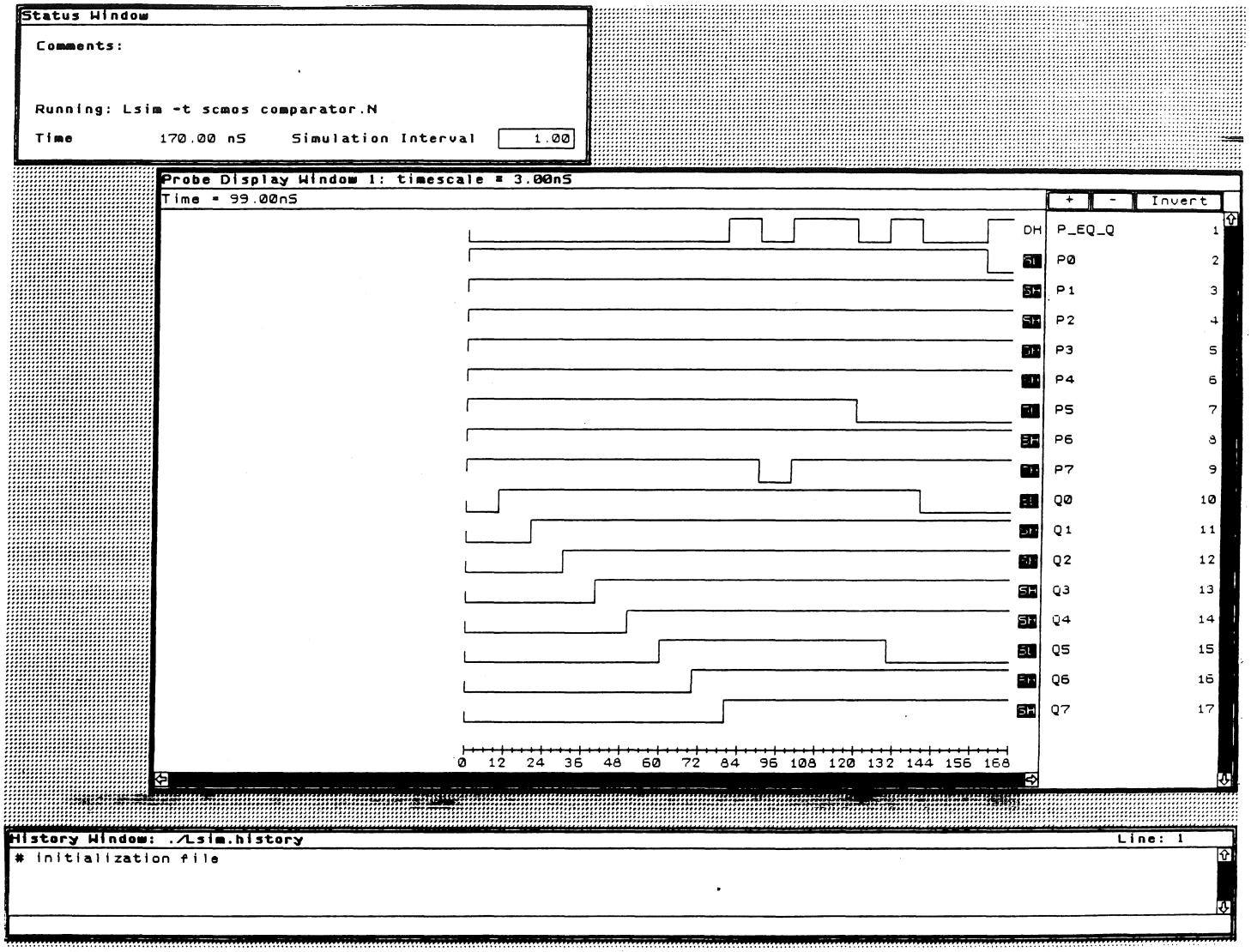

Figure 40: Test Waveform For Comparator

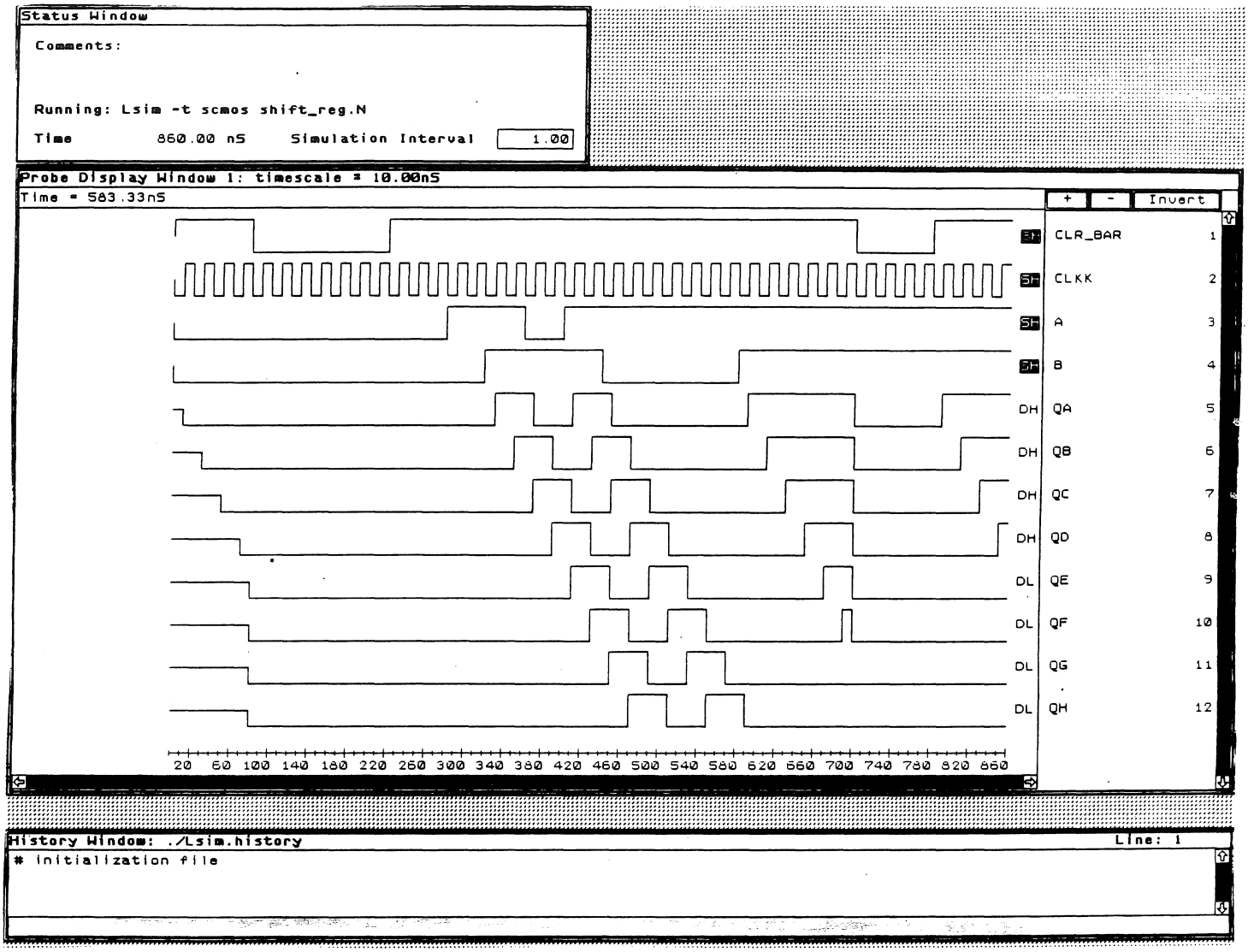

Figure 41: Test Waveform For Shift Register

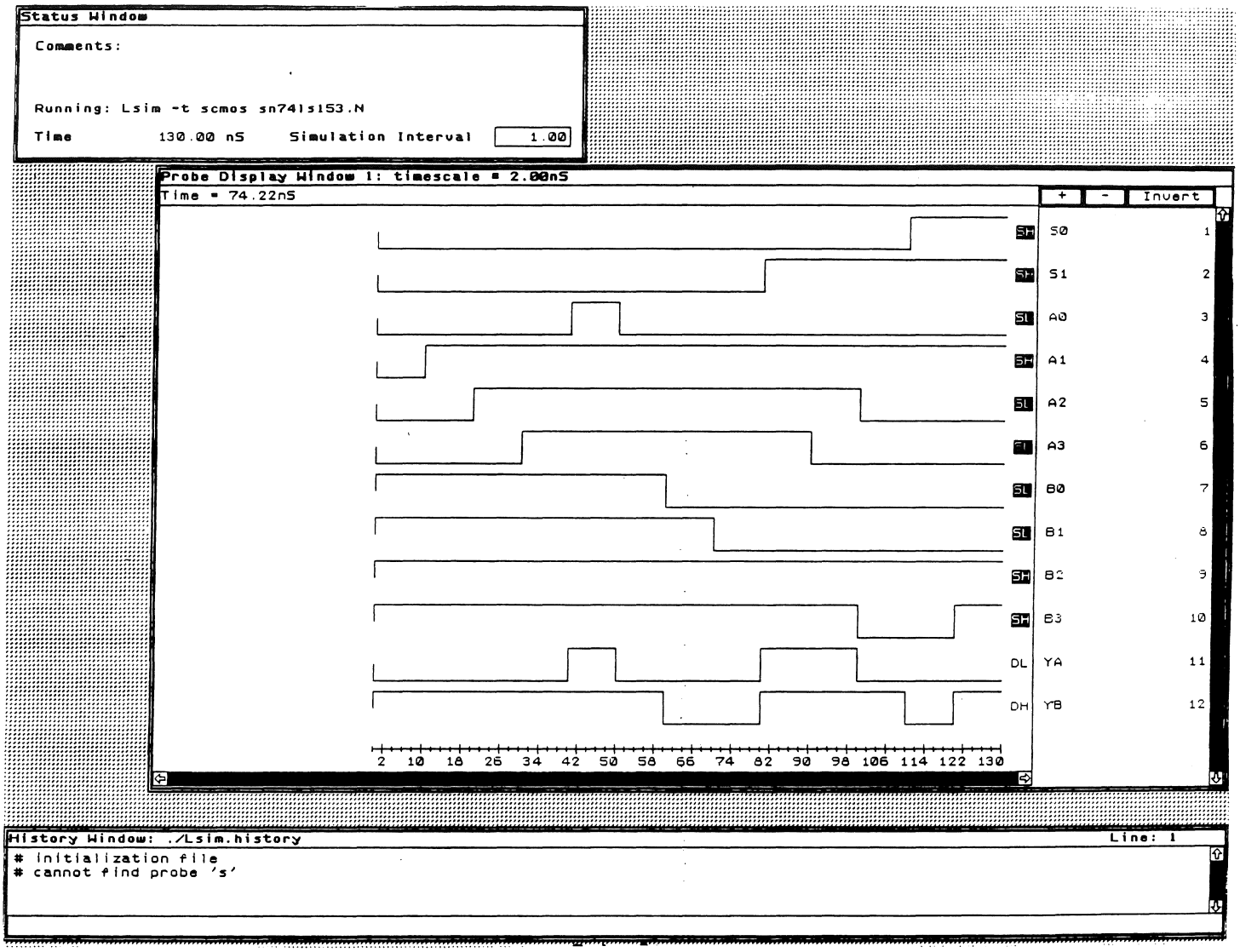

Figure 42: Test Waveform for SN74LS153 (MUX)

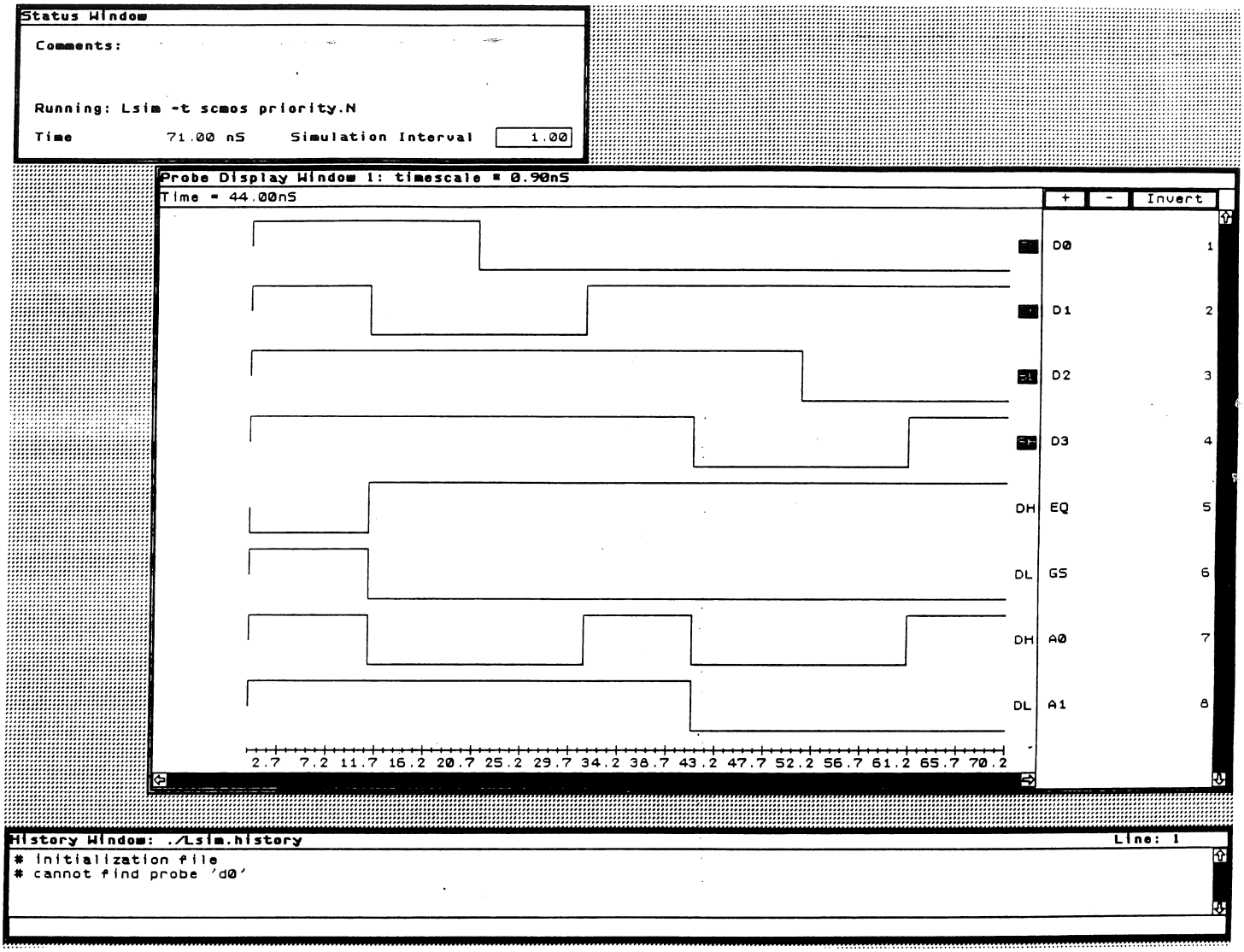

Figure 43: Test Waveform For Priority

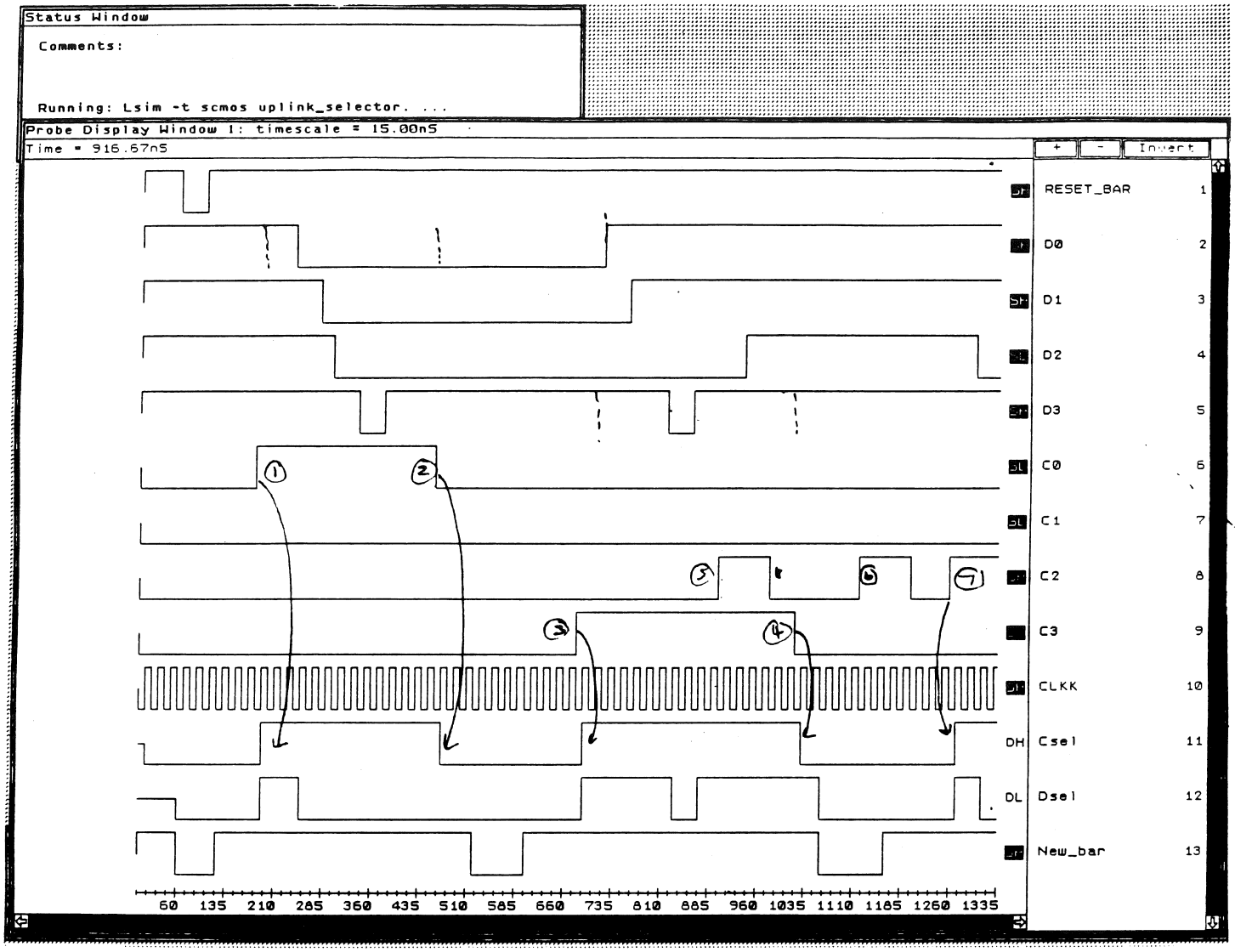

Figure 44: Test Waveform For Uplink Selector

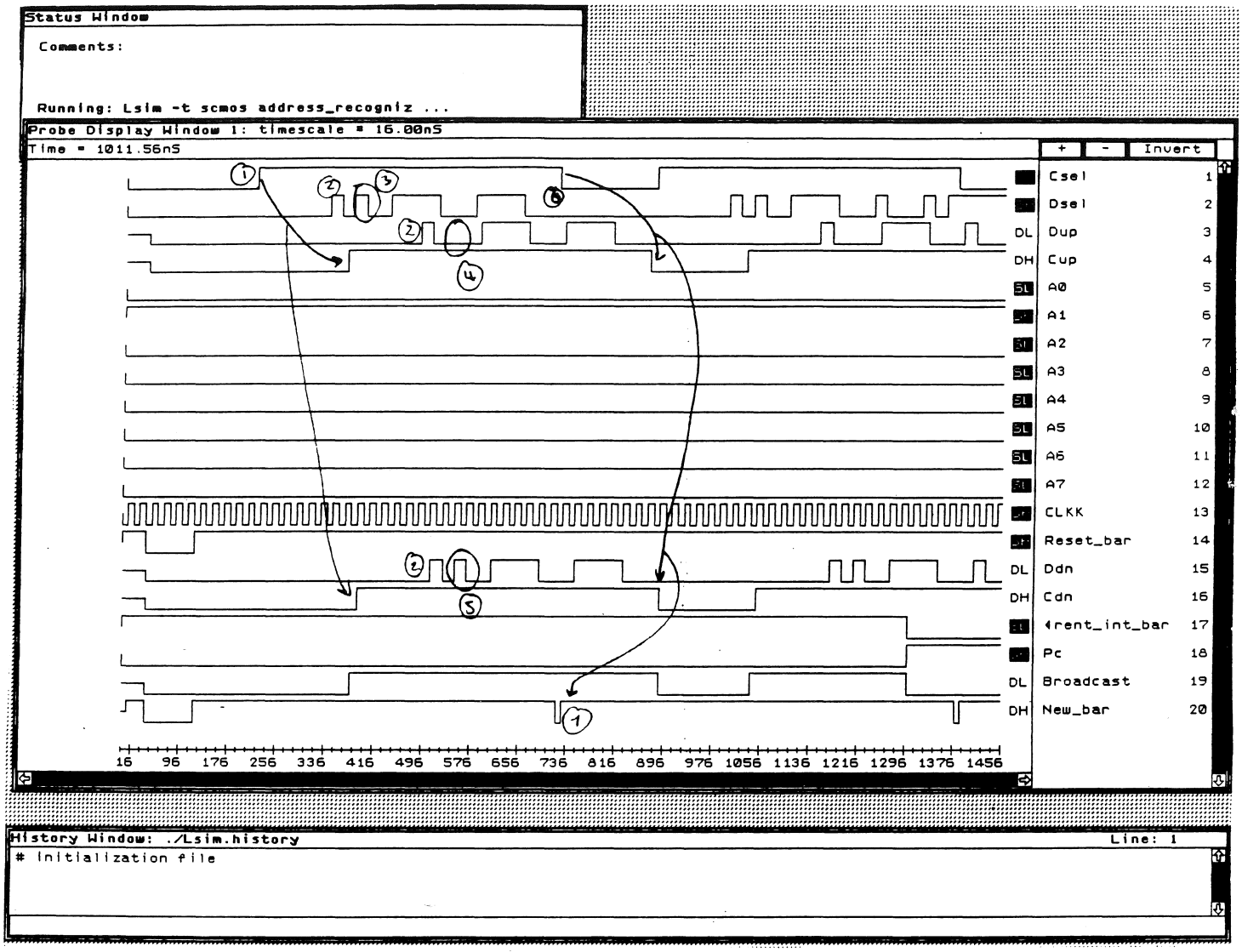

Figure 45: Test Waveform For Address Recognizer

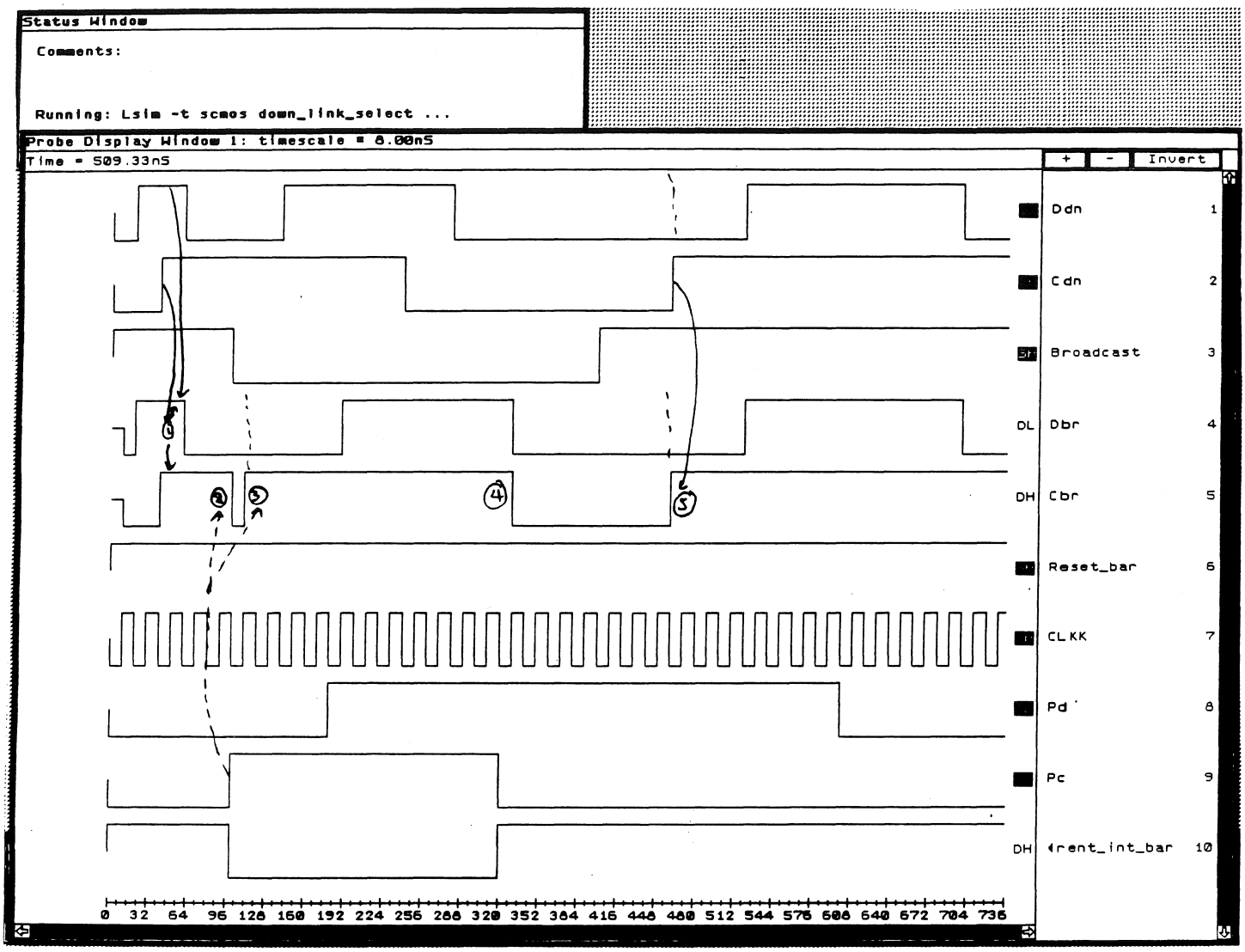

Figure 46: Test Waveform for Downlink Selector

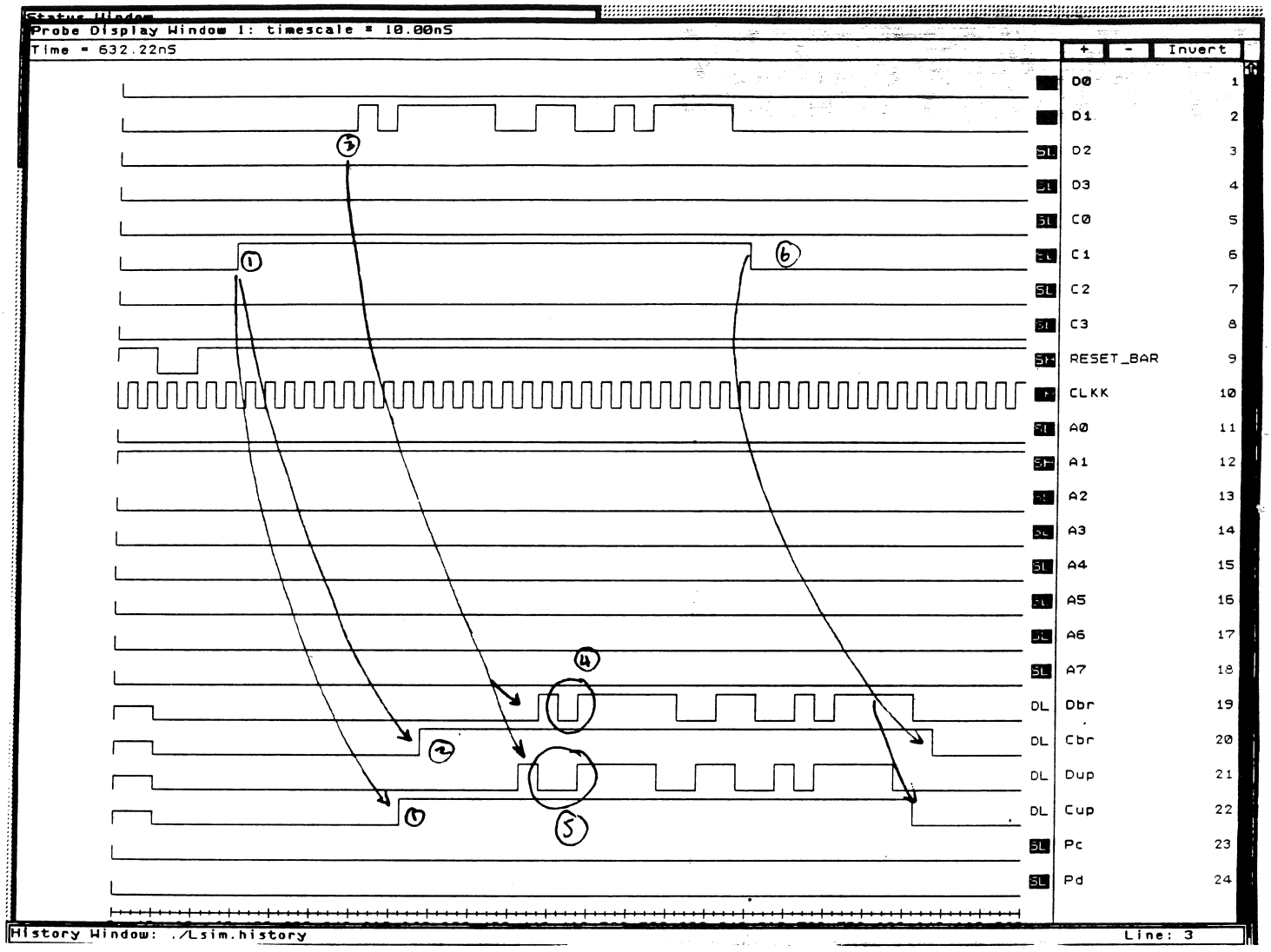

Figure 49: Test Waveform for Test Case 3

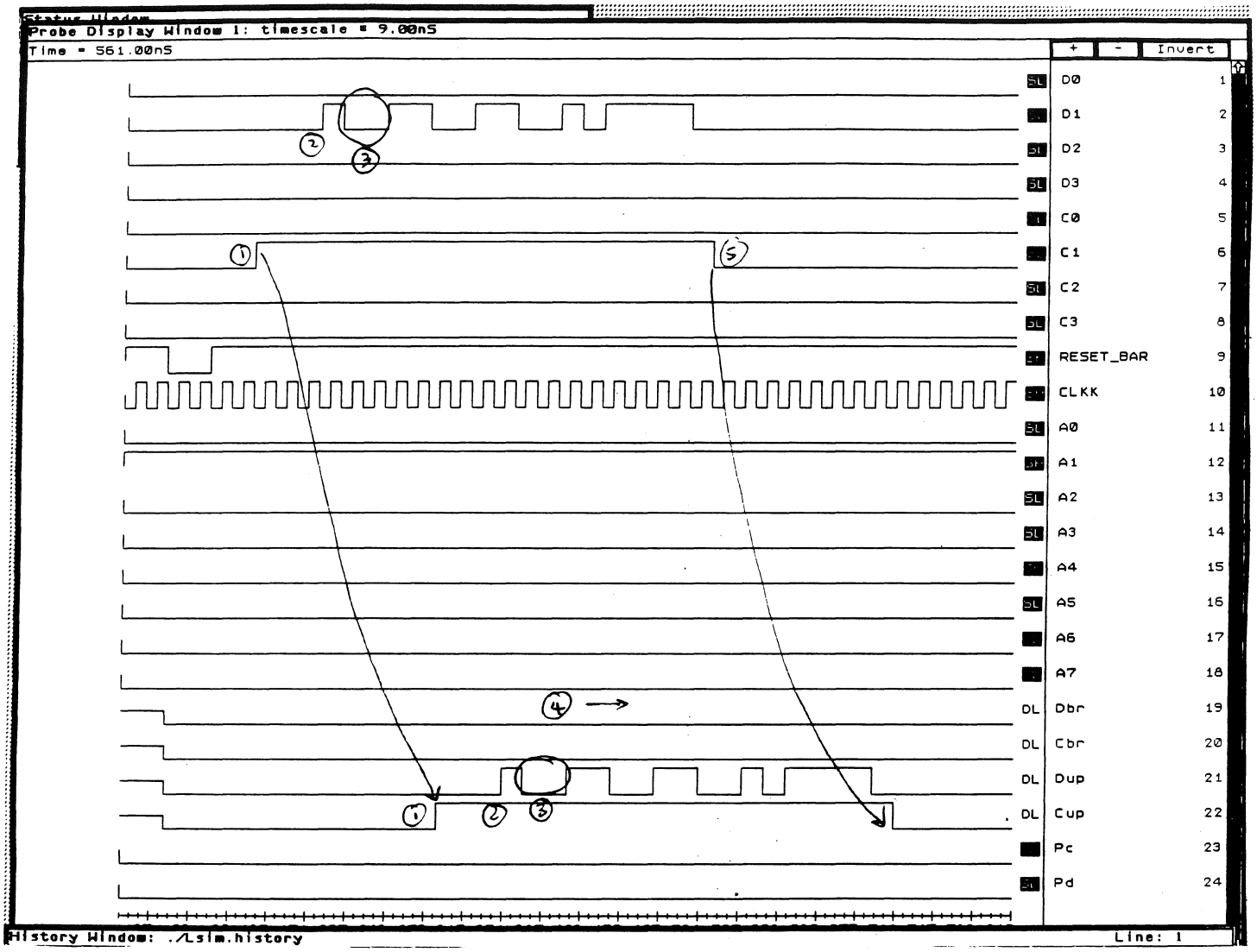

Figure 50: Test Waveform for Test Case 4

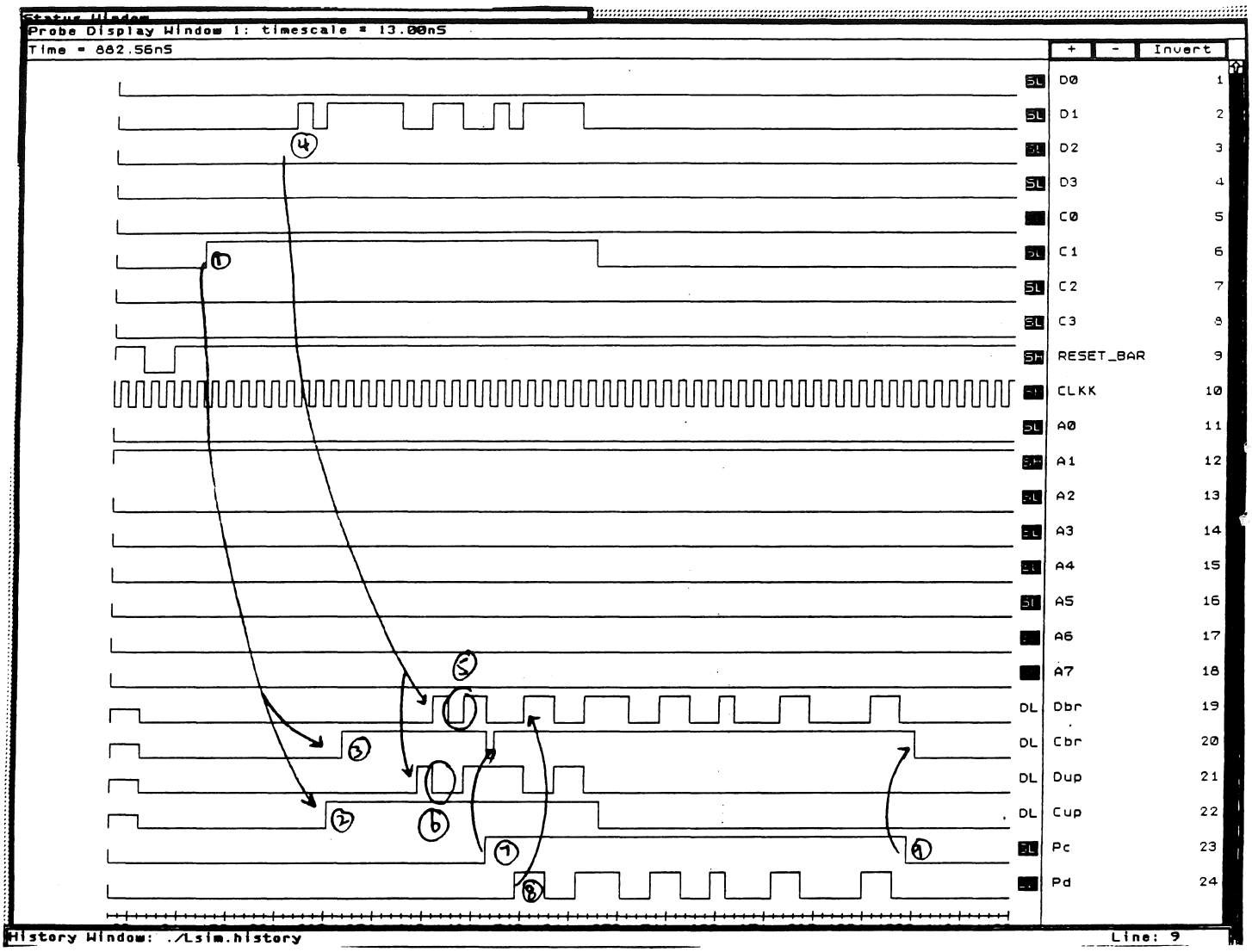

Figure 51: Test Waveform for Test Case 5

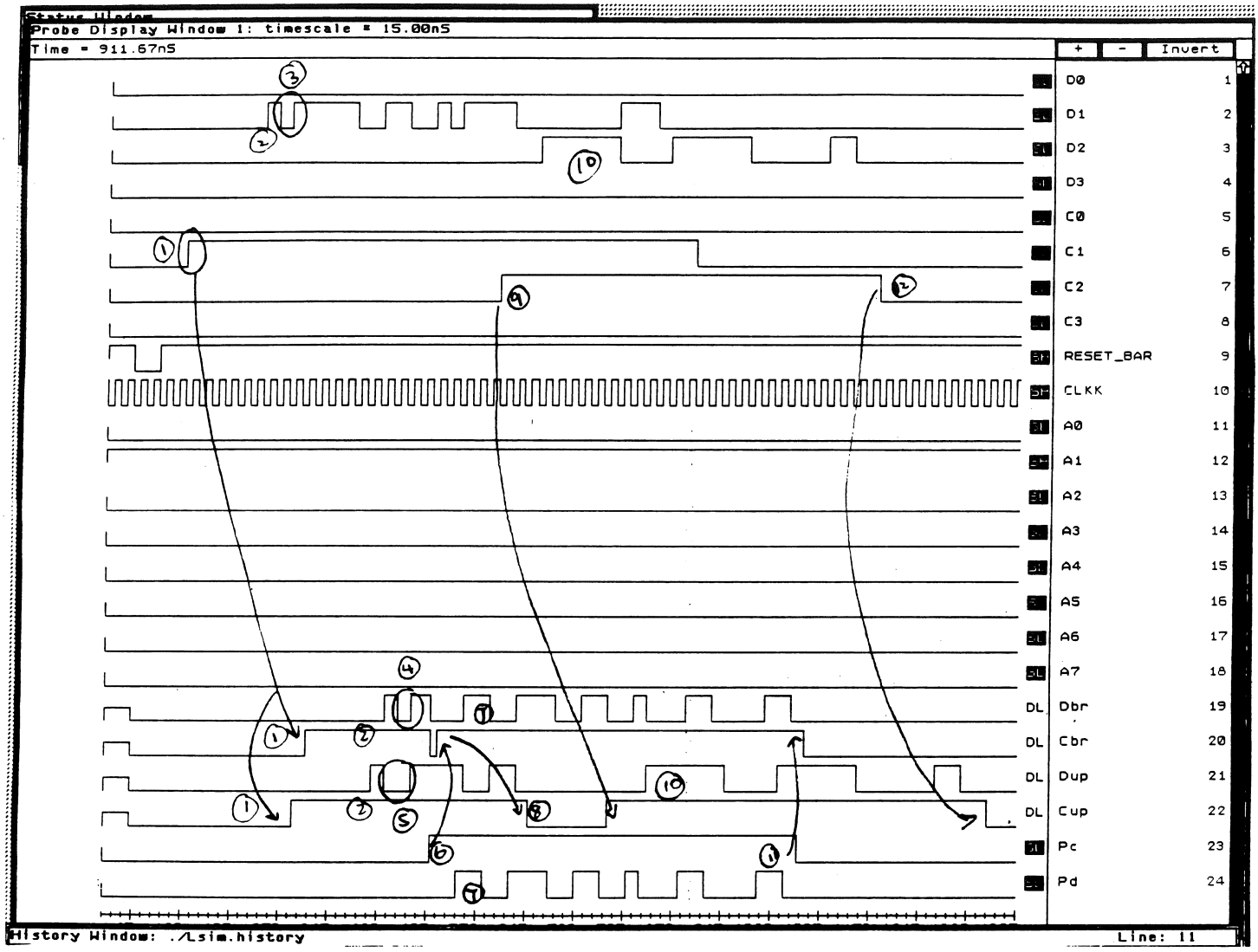

Figure 52: Test Waveform for Test Case 6

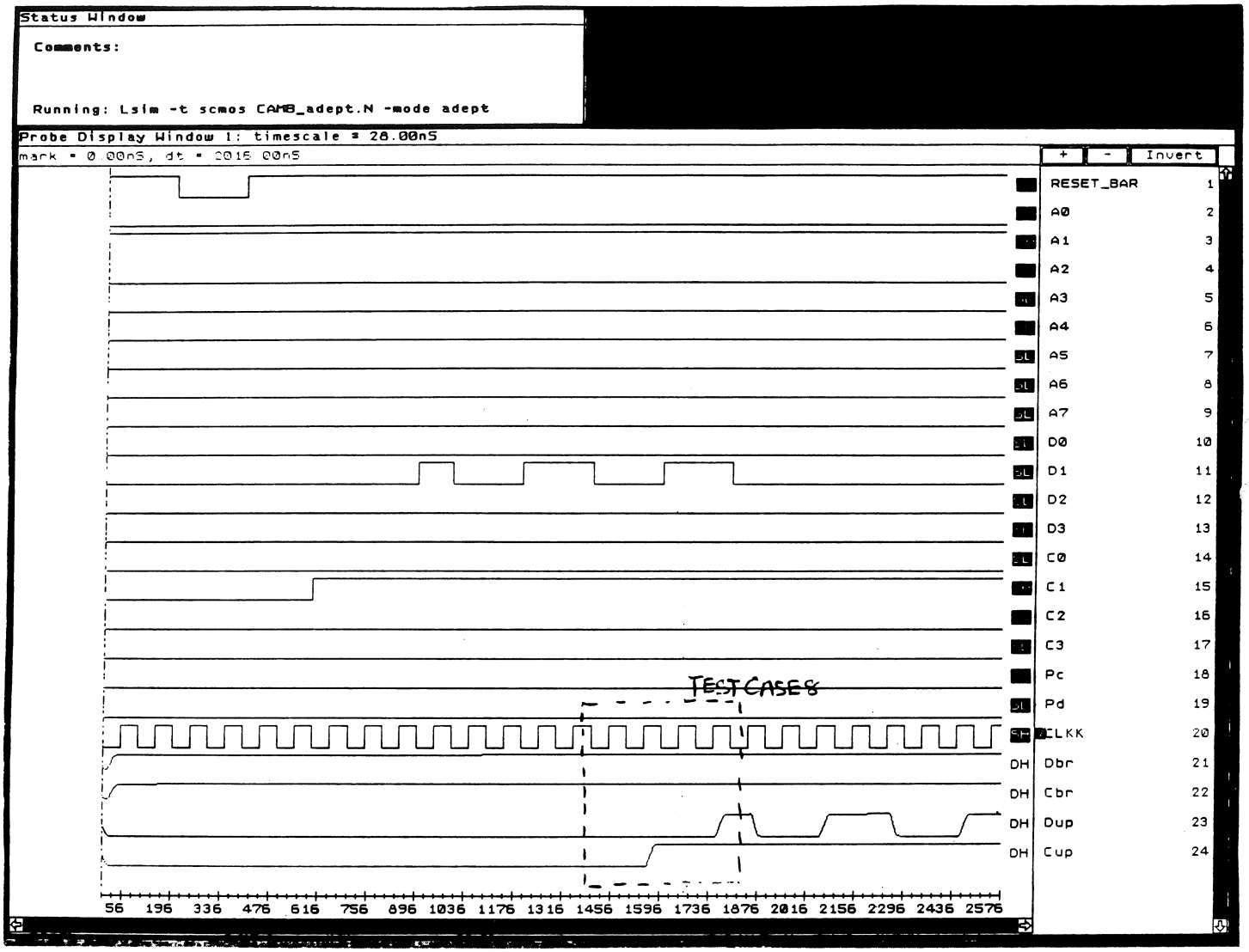

Figure 53: Test Waveform for Test Case 7