AUTOMATIC SCSI ID ACQUISITION

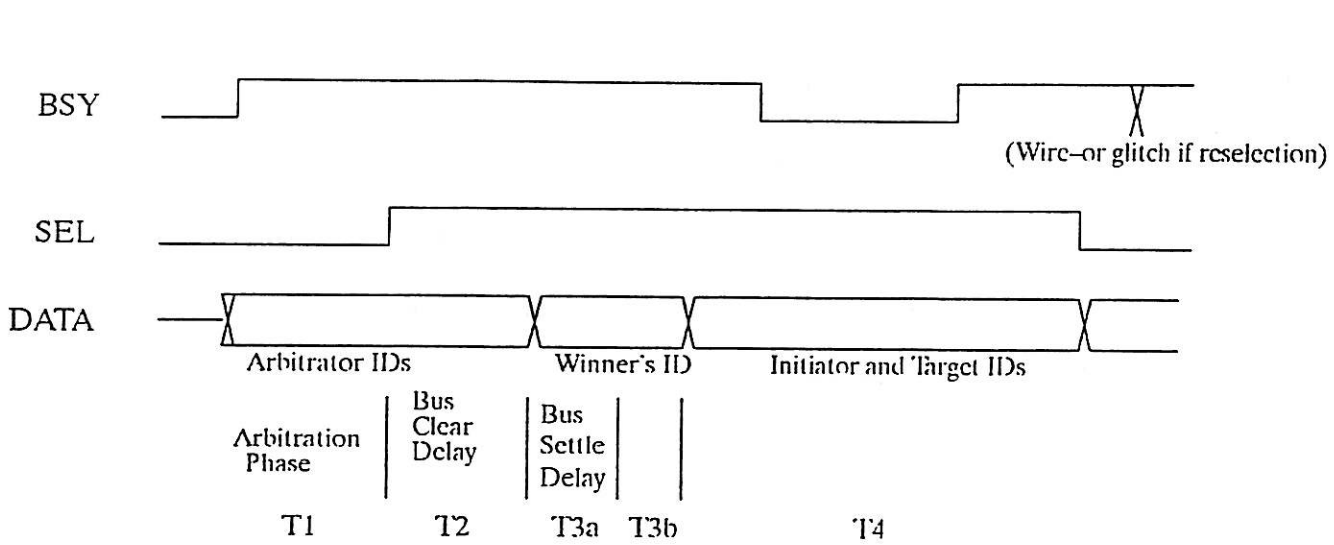* PROBLEM: How does a newly added SCSI Device obtain an unused SCSI ID?

    – Without asserting the RST signal

    – Without disturbing existing SCSI–2 devices on the bus

    – Without seriously impacting existing SCSI chip designs.

    – Without major changes to selection protocol.

* SOLUTION: Make a small change to the selection protocol allowing devices to distinguish between selector and selectee.

    – Provides the capability to "borrow" a SCSI ID for temporary use.

    – May not seriously impact existing silicon. (Opinions on this are needed.)

75

Giles Frazier
IBM AWS Austin

BACKGROUND: (RE)SELECTION PROCEDURE

IBM9966K 3635

BSY

SEL

DATA

(Wire–or glitch if reselection)

Arbitrator IDs   Winner's ID   Initiator and Target IDs

| Arbitration Phase | Bus Clear Delay | Bus Settle Delay | | |
| T1 | T2 | T3a | T3b | T4 |

* Note that during T3b, only the selector's SCSI ID is on the bus.
  (During T3b, the bus has settled and ONLY the selector's ID will be
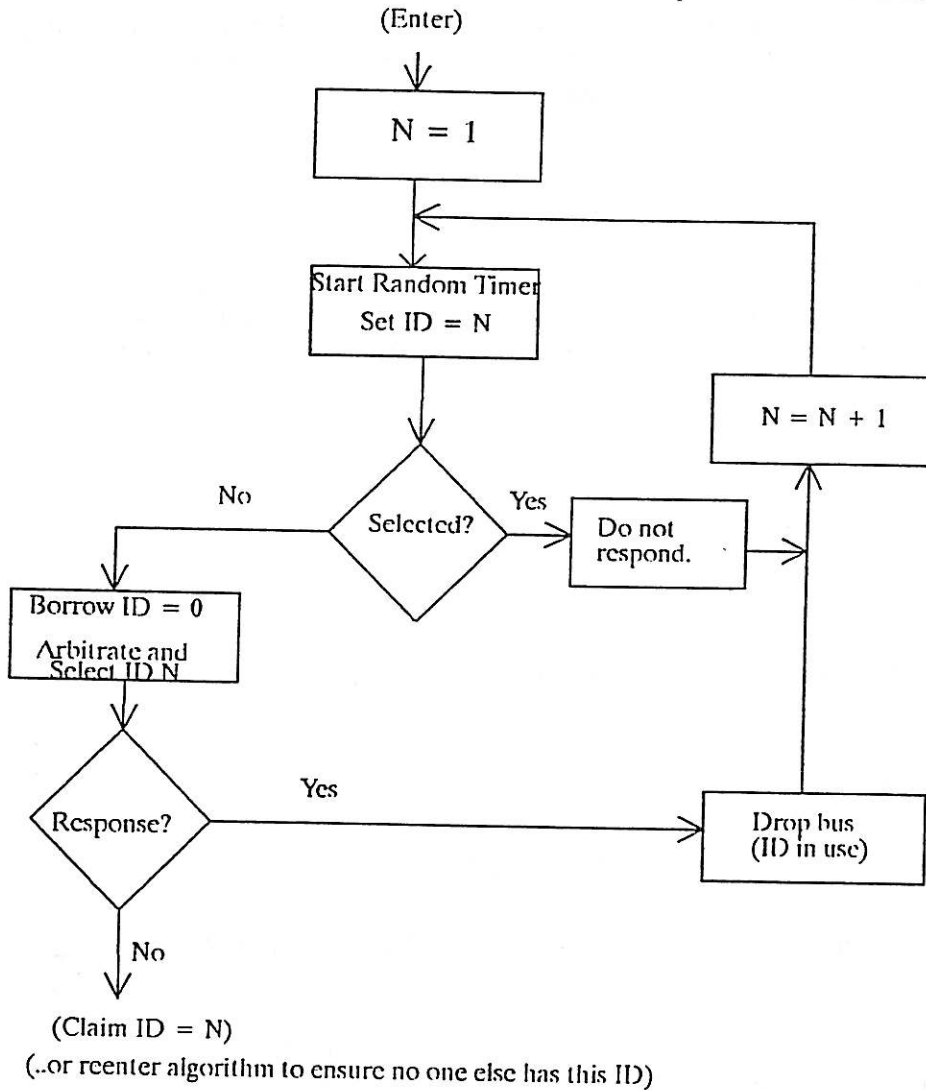 on the bus. The current specification allows T3b to be zero.)

* A device following this new procedure would operate as follows:

   a) When acting as a selector, ensure T3b is TBD ns. (e.g. 400ns)
   b) When acting as a selectee:
      1) Latch the selector's ID during T3b
      2) XOR the Selector's ID with the ID's on the bus during T4
         to determine if device is being selected.

      (Note that the current standard only requires recognition of
      (Own ID) * BSY/ * SEL to determine selection. The selector's
      ID is present but not necessarily recognized.)

Giles Frazier
IBM AWS Austin

WHY MAKE THIS CHANGE?

* The change provides the underlying mechanism for a new device to "borrow" an existing ID.
  - The "borrowed" ID can be used to find an unused ID by polling.
  (e.g. borrow an ID, select another ID, drop bus if it responds.)

* Enables automatic ID selection procedures in which ALL devices select their own IDs. (A simplified automatic ID selection procedure is shown below.)

(Enter)

```
        N = 1

  Start Random Timer
     Set ID = N                              N = N + 1

  No        Selected?    Yes    Do not
                                respond.

  Borrow ID = 0
  Arbitrate and
   Select ID N

          Response?    Yes              Drop bus
                                        (ID in use)

             No

  (Claim ID = N)
  (..or reenter algorithm to ensure no one else has this ID)
```

Giles Frazier
IBM AWS Austin

WHAT IS NEEDED TO ALLOW THIS?

Two changes to SCSI (Re)Selection Procedure

- All devices must latch the winning arbitrator's SCSI ID at least a Bus Clear Delay following the time when BSY and SEL become true. (i.e. during T3b above)

- A device determines that it is (re)selected when the SEL (and I/O) signals *and the XOR of it's SCSI ID and that of the arbitrator's* are true and the BSY signal is false... (*italics* indicate a change)
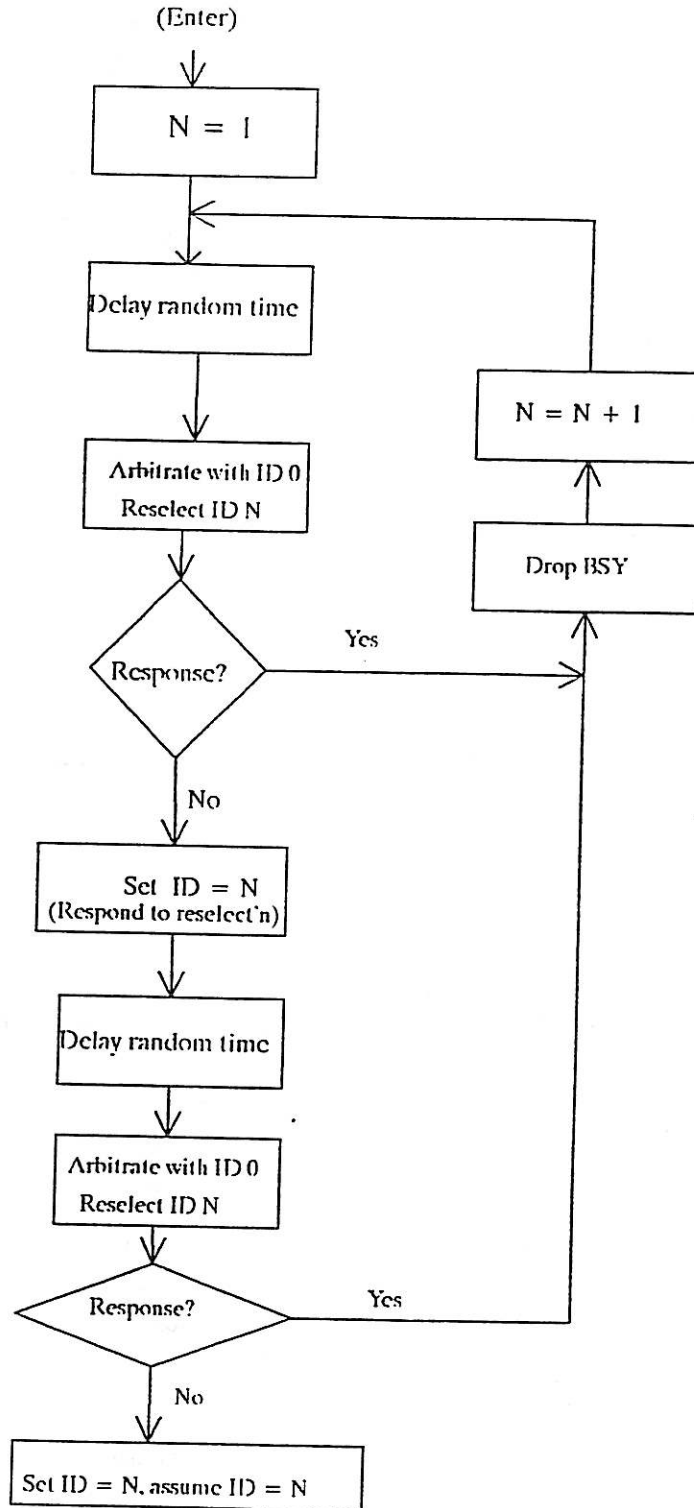
One Change to the Arbitration Procedure

- A winning device now has to delay a Bus Clear Delay before asserting the (re)selectee's address. The winning device would have to delay a Bus Clear Delay plus some TBD (e.g. 400ns) before asserting the (re)selectee's address.

ADVANTAGES

* Automatic ID selection schemes are made possible. (Current proposals could use this scheme.)

* No RST is required

* Devices following this procedure can coexist with existing devices.

* Potentially minor impact to existing chip designs. (Further evaluation required.)

DISADVANTAGE: IMPACT ON SILICON
(Consider the algorithm on the next page for an alternative)

Giles Frazier
IBM AWS Austin

(ID Borrowing without Silicon Impacts?)

(Enter)

↓

```
┌─────────────────────┐
│        N = 1        │
└─────────────────────┘
            │
            ↓
┌─────────────────────┐
│  Delay random time  │
└─────────────────────┘
            │
            ↓
┌─────────────────────┐
│  Arbitrate with ID 0│
│    Reselect ID N    │
└─────────────────────┘
            │
            ↓
         ◇ Response? ◇ ──── Yes ────→
            │ No
            ↓
┌─────────────────────┐
│     Set  ID = N     │
│ (Respond to reselect'n) │
└─────────────────────┘
            │
            ↓
┌─────────────────────┐
│  Delay random time  │
└─────────────────────┘
            │
            ↓
┌─────────────────────┐
│ Arbitrate with ID 0 │
│    Reselect ID N    │
└─────────────────────┘
            │
            ↓
         ◇ Response? ◇ ──── Yes ────→
            │ No
            ↓
┌──────────────────────────┐
│ Set ID = N, assume ID = N│
└──────────────────────────┘
```

```
┌─────────────────────┐
│     N = N + 1       │
└─────────────────────┘
            ↑
┌─────────────────────┐
│      Drop BSY       │
└─────────────────────┘
```

(The bottom 4 steps could be repeated multiple times to attain a great level of certainty that no two devices used the same ID.)