1. Overview

- Models the behavior of a Serial Bus Net with arbitrary number of nodes and buses.
- An input script creates the modeled environment and applies events to the model.
- Simulation result is recorded in a trace file.
- Simulator written in Visual C++, executes as a Windows application. It is intended to be a tool internal to Sony with minimal user interface.

1.1. Purpose

To provide Sony Communication Architecture group a testing tool for P1394 bridges.

1.2. Scope

■ Implemented:

- Asynchronous packet transmit/receive, all types of transactions are implemented except LOCK LITTLE_ADD, LOCK BOUNDED_ADD and LOCK WRAP_ADD.
- Routing of asynchronous packet
- Isochronous packet transmit/receive
- Packet transmission time on bus (1 simulation cycle equals approx. 1280 nSec)
- Bus reset on node connect/disconnect
- Net setup procedure with multiple bridge manager capable nodes contending for the ownership of the net.
- Bridge manager heartbeat

■ Plan to implement:

- Routing of Isoc packet
- Reset notification
- Busy retry protocol
- Fair arbitration, urgent arbitration
- Multiple data transmission rate

■ Will not implement:

- Bit level data movement
- Electrical aspects
- High level protocols

1.3. An input file example

```
// TEST.9 - 4 Nodes, 4 buses, 2 bridges

INCLUDE TEST.ALL
OUTPUT out.9
bus   new=B0
node  new=N00 connect = B0  brmgr
bus   new=b1
node  new=N10 connect = b1  brmgr
bus   new=b2
node  new=N20 connect = b2  brmgr
bus   new=b3
node  new=N30 connect = b3

bridge new=bx portal=3
PORTAL name=bX_p0 connect=b0
PORTAL name=bX_p1 connect=b1
PORTAL name=bX_p2 connect=b3

bridge new=by portal=2
PORTAL name=by_p0 connect=b1
PORTAL name=by_p1 connect=b2
ISOC NAME=N00 TALK=START CHAN=3 len=0x20
ISOC NAME=N10 LISTEN=START CHAN=3
delay 1000

ISOC NAME=N10 LISTEN=STOP CHAN=3

Xmit src=N20 dest=N00 subaction=LOCK FETCH_ADD size=QUADLET
data=0x12345678 addr=0xfffff0000550

xmit src=N20 dest=BX_P1 subaction=LOCK MASK_SWAP size=OCTLET
data=0x3312345678 ARG=0xFFfff0000 addr=0xfffff0000550

xmit src=N10 dest=BY_P0 subaction=wr_BLK BLKDATA=fffff0000604
addr=0xfffff0000550

xmit src=N10 dest=N30 subaction=wr_BLK BLKDATA=fffff0000604
addr=0xfffff0000550

delay 50
Trace BRMGR
delay 2
```

1.4. An output file example

```
--------------------------------
Cycle #3184   ISOC Cycle Tick=47
....Printing all nodes..
  Node=BY_P1, ID=0/1 EUI=BBBBBBBB 9
  Node=BY_P0, ID=1/2 EUI=BBBBBBBB 8
  Node=BX_P2, ID=3/1 EUI=BBBBBBBB 7
  Node=BX_P1, ID=1/1 EUI=BBBBBBBB 6
  Node=BX_P0, ID=2/1 EUI=BBBBBBBB 5
  Node=N30, ID=3/0 EUI=BBBBBBBB 4
```

```
  Node=N20, ID=0/0 EUI=BBBBBBBB 3
  Node=N10, ID=1/0 EUI=BBBBBBBB 2
  Node=N00, ID=2/0 EUI=BBBBBBBB 1
   RecvFifo  SrcBus SrcID  DestBus DestID   Tl     TCode    Rcode     ExtTCode
               0      1       1      0      20   LOCK_RSP RSP_COMPLETE
   004050B0
   00010000
   00000000
   00040000
   00000000
   FFFF0000
   EEEEEEEE
               0      1       1      0      21   LOCK_RSP RSP_COMPLETE
   004054B0
   00010000
   00000000
   00040000
   00000000
   00000002
   EEEEEEEE

....Printing all portals..
  Portal=BY_P0,  Ordinal=0  BusID=1  Low=1  Up=3  Rte=EXC BrMgr=0/0
owner=BBBBBBBB 3
  Portal=BY_P1,  Ordinal=1  BusID=0  Low=1  Up=3  Rte=INC BrMgr=0/0
Owner=BBBBBBBB 3
  Portal=BX_P0,  Ordinal=0  BusID=2  Low=2  Up=2  Rte=EXC BrMgr=0/0
Owner=BBBBBBBB 3
  Portal=BX_P1,  Ordinal=1  BusID=1  Low=2  Up=3  Rte=INC BrMgr=0/0
Owner=BBBBBBBB 3
  Portal=BX_P2,  Ordinal=2  BusID=3  Low=3  Up=3  Rte=EXC BrMgr=0/0
Owner=BBBBBBBB 3

....Printing BrMgr & Net Topology Map..
  BrMgr=N20
Bus PhyID Type  Num_Portal Active Rte   LO   Up  NextPortalInBr   BrMgrCapable
  0   0   NODE      0        0          3FF  3FF     -1  -1            1
  0   1   PORT      2        1    INC     1    3      1   2            0
  1   2   PORT      2        1    EXC     1    3      0   1            0
  1   0   NODE      0        0          3FF  3FF     -1  -1            1
  1   1   PORT      3        1    INC     2    3      2   1            0
  2   1   PORT      3        1    EXC     2    2      3   1            0
  2   0   NODE      0        0          3FF  3FF     -1  -1            1
  3   1   PORT      3        1    EXC     3    3      1   1            0
  3   0   NODE      0        0          3FF  3FF     -1  -1            0

  Orphan Nodes:
--------------------------------
Cycle #3185   ISOC Cycle Tick=46
```

2. Simulation Result Findings and Discussions

2.1. Bridge Portal Register Locations

- PORTAL_SELECT moved to outside of the Portal-Specific window to avoid "nested windowing" effect.

- A new register added - "PORTAL_NODEIDS". This register is an exact copy of the NODEIDS register, reading from it gets a copy of the node's NODEIDS register. Writing to PORTAL_NODEIDS will cause its new register content to be copied over to the node's NODEIDS register. During Net setup, a bridge manager competing node enumerates buses through writing to NODEIDS.BUSID of all nodes. For "INDIRECT" portals, namely, portals a bridge manager competing node can have

access to only through portal-specific window, since the NODEIDS register is not inside the window, to assign a bus ID to an "INDIRECT" portal can only be done with the introduction of "PORTAL_NODEIIDS" register.

- 2 bits added to PORTAL_CONTROL register.
  1) READY, Set by the bridge when it is done setting up the portal on power up.
  2) ACTIVE, On power up, when a bridge detects more than one of its portals are connected to the same bus, it will enable only one of these portals by setting its ACTIVE bit to 1 to break the loop connection, and leave the rest of them in non-active state.

- 1 bit added to bridge manager control and status registers.
  BRMGR_COMPETING - Set by a node itself while it is setting up the net, reset when it is done setting up the net, or when it detects other contender has winning EUI. It is also reset by other nodes with winning EUI to inform this node that it has lost the contention. Each bridge manager contending node checks this bit periodically while setting up the net.

  When multiple bridge manager capable nodes contending to setup a net, if collision happens inside a bridge portal (Comparing portal OWNER_EUI_64 with its own EUI), the losing node will withdraw from the competition immediately. When a node with winning EUI has reached a bus in which a bridge manager capable node with a losing EUI is residing, and the node with the losing EUI is in the process of configuring the net through a portal other than the one the winning node came in through, it becomes more difficult for the losing node to know right away that it has lost the contest. If at this point, the winning node simply writes to all nodes in bus their new bus ID without doing any other thing, the losing node will continue setting up the net with its new bus ID. In order to have the losing node withdraw from the contest right away, the winning node resets the losing node's BRMGR_COMPETING bit, when the losing node sees that the bit is reset by others, it quits the net setup procedure.

- Need to have an interlock mechanism to prevent simultaneous access to PORTAL_SELECT register to prevent inadvertent access to incorrect bridge portal registers?

- Should we group "rte" field with the portal routing bounds into the same register to reduce net traffic during net setup?

2.2. Net Setup Procedure

- There is no longer a separate procedure to determine who is going to be the bridge manager. All bridge manager capable nodes start contending for the role of the bridge manager the minute they see any local bridge freed.

- When a contending bridge manager assigns bus IDs to all nodes in a bus, instead of broadcasting quadlet write transaction to NODE_IDS register, LOCK MASK_SWAP transaction is issued to each individual nodes in bus. The reason being that in a backplane environment, a write to the phy ID field will be stored rather than ignored. Also, since there is no response packets for broadcast transactions, a bridge manager node will have no guarantee that all nodes in the bus have received the broadcast write.