1394 Follow-On Study Group, Bridge Concentration
Minutes of meeting 2/14/96 at Sony in San Jose

Opening

Dick Scheel, temporary Chair, called the meeting to order. He announced that this was a meeting of the IEEE 1394 follow-on study group addressing the topic of bridging. Since a PAR has not yet been approved (or even written), the group is not yet an IEEE "working group". It was also stated that no votes were intended, since this was not yet a formal meeting under IEEE rules (in particular, the meeting was held with slightly less than 30 days notice). Although this was not a formal IEEE meeting, it was intended to follow IEEE procedures as much as possible. Most importantly, the meeting was open to any interested party.

The following people attended the meeting:

Dave James	Apple	408-974-1321	dvj@apple.com
Peter Johansson	Congruent Software	510-531-5472	pjohansson@aol.com
Nobuo Furuya	NEC	+81-44-435-1616	n-furuya@lsi.tmg.nec.co.jp
Kaz Abe	NEC Systems Lab	415-528-5904	kabe@SJ-PCEG.ccgw.nec.com
Ron Laborde	PACT	+44-117-970-7168	ron@pact.srf.ac.uk
Arshad Hasan	Samsung Info Systems	408-434-5518	ahasan@sisa.samsung.com
Ed Carmona	Seagate	408-456-4082	ed.carmona@conner.com
Horatio Lo	Seagate	408-456-4216	Horatio.Lo@conner.com
Andy Jones	SGS-THOMSON	+44-117-970-7181	andyj@bristol.st.com
Colin Whitby-Strevens	SGS-THOMSON	+44-1454-611-500	colinsw@bristol.st.com
Dick Scheel	Sony	408-955-4295	dicks@lsi.sel.sony.com
Bill VanLoo	Sun Microsystems	415-786-6870	Bill.Vanloo@Eng.Sun.COM
Alan Wetzel	Texas Instruments	214-442-2663	awetzel@ti.com

The following agenda was presented:

1. Presentation by Peter Johansson on bridge design work in progress
2. Discussion of PAR
3. VESA Home Network committee information
4. Possible discussion of "switching fabric" ideas, if time is available

Peter Johansonn's presentation

Peter distributed copies of his presentation titled "Serial Bus to Serial Bus bridges". He then presented the material, with the following discussion.

1. Should there be one EUI-64 per bridge, or one per portal? For now, this will be kept open. As the bridge design develops, we will see how this affects the design. It was pointed out that the same needs to be examined for the entire configuration ROM, as well as the cycle clock. There is value in keeping the shared portion of the bridge as stateless as possible.
2. Must a packet from bus A to bus B follow the reverse of the path of a packet from bus B to bus A? For now, nothing has been seen in the algorithms being worked on that would be affected by either answer to this question. One possible benefit to keeping the paths the same is that it makes it easier to debug broken request/response pairs. Should there be any need for a bridge to be aware of whether a packet is a request or a response?

3. It was proposed that perhaps a minimum requirement of a bridge be the ability to propagate the isochronous cycle clocking, but not necessarily the ability to route isochronous packets.
4. Is it important that all bridges have the same number of cycles delay in forwarding isochronous packets? Probably not, just that the number be constant for each bridge. Additional questions arose: Must the constant be the same for all pairs of portals in the bridge, and must the constant be published (perhaps in the configuration ROM)?
5. Remote transactions were discussed. This term refers to the bridge function that allows transactions addressed to one portal of a bridge to cause another portal of the bridge to initiate a specified transaction and return the response. This permits initiating transactions on buses that have not yet been configured during network setup.
6. Async routing was discussed, including the sufficiency of the base/bounds model to provide minimum (though generally not optimal) routing capability for any network of buses.
7. If a portal is disabled for isochronous cycle clock propagation, must it also be disabled for isochronous packet routing? Phrased differently, must an isochronous packet follow only the cycle clock tree, or can it take a shorter path between branches? The general opinion was that we are not sure, but it seems like any path is acceptable. It was noted that since the path of an isochronous stream across the network is set up by an application, then any application that may need the path to follow the clock distribution tree may chose to set up that path. Most likely the standard will not require any particular path.
8. There was discussion about bridges needing to read and/or alter any of the data portion of packets routed through the bridge. The current known example is time stamp information in MPEG streams. Dick took the action item to get information about what is the range of possible data modifications bridges may have to perform.
9. Next Peter described the use of Plug Control Registers for isochronous routing. (It was noted that the acronym PCR is used in MPEG for “Program Clock Reference”, so perhaps the 1394 community might want to use the term “plugs” rather than “PCR”.)
    a) How will we handle limited bandwidth and channel number resources within the bridge switch fabric?
    b) In the PCR model a node publishes the number of PCR’s it has, so there is a limit of a fixed number of plugs. This does not address bandwidth. A bridge might be able to handle more 1 Mbps streams that 15 Mbps streams.
    c) One possible solution is for the switch fabric to have a bandwidth allocation register just like a 1394 bus.
    d) A more capable solution is to have some defined request/response where an application can ask for a connection through a bridge with a given bandwidth (and channel allocation?). The response would indicate whether the requested connection was successfully set up.
    e) If the request/response method is used, should the function appear to be attached to the portals or to the shared part of the bridge? If connected to the portals, should it actually be just a per-portal view of a common function?
10. Async transaction integrity was discussed next. This addresses the problem of resetting of a bus in a network causing nodes to change their phy_id, while a packet is in transit through the network addressed to the now invalid node_id.
    a) Peter presented a model using two registers in each portal:
        i) a 64 bit “outbound phy_id” blocking register
        ii) a 1024 bit “inbound bus_id” blocking register
    b) After Peter presented the proposed algorithm, several attendees indicated that it seemed overly complex - might there not be a simpler (and therefore more likely to be correct) model?
    c) Several partial ideas were discussed, involving fewer bits plus bridge manager involvement, or using some form of timeout. None of the ideas seemed to be complete enough.
    d) It appears that all “remote transaction capable” nodes (nodes that can initiate a transaction to a node on another bus) must reset there own bus_id to 3FF on any bus reset. This is the only way to avoid mis-routed packets when two buses in a network are plugged together. It is also necessary, and is probably an oversight in the original

1394-1995 document, that no node may initiate a transaction to another bus if their source bus_id = 3FF (this would prevent returning a response packet). Note that resetting the bus_id on bus reset is a change from the 1394-1995 document. Since no known remote transaction capable devices have ever been built, we felt it was safe to add this behavior for this new class of devices. However, this should be published for comment - there may be some negative effect that this group did not see.

e) It is important that all nodes in a network receive notification when any bus in the network is reset.

f) This subject clearly needs more work. Dick, Peter, and Dave James indicated interest in pursuing this.

11. The discussion returned to remote (proxy) transaction capability.
    a) There are two ways to implement this:
        i) Set up the transaction information with writes to registers in the bridge, and then issue a transaction to the bridge that causes the proxy transaction to be sent.
        ii) Set up some mapping registers in the bridge, and from then on all transactions addressed to a certain place automatically are mapped and forwarded out the other portal. Dave James will write a brief description of how this might be defined.
    b) Each method has pros and cons for the hardware and software.
    c) Should any transaction be permitted, or only a minimal subset? Which is easier/better? Possible subset ideas:
        i) async only
        ii) quadlet aligned
        iii) Qread, Qwrite, 64 bit compare/swap minimum required transactions
        iv) are block read and block write needed in some cases?
12. Peter presented the section on bus enumeration and network setup
    a) Dave James recommended that rather than each bridge storing several items of information about the bridge manager, it should only store a 64 bit address of a data structure in the bridge manager that contains all the needed information. This way the bridge manager can make atomic changes to multiple data items simply by changing the pointer. This data structure would probably hold the EUI-64 of the bridge manager along with information that would be used to arbitrate between contending bridge manager capable nodes, and heartbeat information.
    b) Heartbeat function:
        i) Is it sufficient to only do heartbeats between bridge manager capable nodes, and not include the bridges?
        ii) A write is better than a read as a heartbeat transaction, since it is a unified rather than split transaction. Maybe the transaction is a write of the pointer to the bridge manager data structure mentioned above.
        iii) During network setup, there may be times when some areas are up and running while the bridge manager is very busy setting up other areas of the network. While the manager is busy, it may be too difficult to keep the heartbeat function running. Possible solutions during this busy time:
            a) Start out with other nodes reading from bridge manager, switch to bridge manager writes after setup is complete.
            b) The bridge manager writes the heartbeat to one “proxy” heartbeat originator (probably another bridge manager capable node?), and the proxy sends the heartbeat to all other nodes that need the heartbeat.
13. Action item (for whom?): Make sure 1394-1995 specifies that all internal queues/FIFOs (up through the transaction layer) are cleared on bus reset.
14. “Unresolved Issues”
    a) For now the idea of a “device model” is withdrawn, since the new ideas brought up at this meeting may remove the need for this functionality. The reservation of bandwidth in the switching fabric may be best handled with this model, though.
    b) The contents of the NODE_IDS register after reset has already been addressed at this meeting.

PARs

The group discussed how the PARs for the 1394 follow on work should be organized. Peter had distributed email suggesting a different division of tasks than was selected at the study group meeting in Dallas in January. The best this meeting could come up with was:

- 1394.2 - Isochronous PCRs, 1394-1394 bridges, protection & security, async streams
- 1394.? - Backplane phy/link: errata, isochronous
- 1394.1 - DS-encoded cable phy/link: arbitrated fast reset, A/V cable, incremental reconfig, loop breaking, phy/link integration, distance > 25 meters
- 1394.3 - (link) Insertion buffer protocols, >25 meters, >800 Mbps, efficient
- 1394.4 - P0 <note - I think this meant physical layer 0, of multiple possible new phys>
- 1394.? - P1 <note - I think this meant physical layer 1, of multiple possible new phys>
- **. . .**
- 1394.? - wireless

The actual number that follows the “1394.” Is not up to us, it’s up to the IEEE committee that approves the PARs. Probably the only one where we care much about the number is that “.1” be the document that gives errata to the original document. No editor has been chosen for that yet. Al Wetzel said he would be willing to at least write the PAR.

Peter Johansson and Dick Scheel will work on the initial draft of the PAR that addresses bridging. Before the draft is sent to the reflector, the following people volunteered to review it:

Colin Whitby-Strevens
Dave James
Peter Johansson
Dick Scheel
Al Wetzel