

Contained in this package:

1. AMBSI1 & AMBSI2 Design Review Meeting Report
2. AMBSI1 & AMBSI2 requirements compliance matrix supplied to the review panel.
3. Received comments and replies.

DRAFT ALMA Review - AMBSI Design, Review Report
Meeting held 2000-Mar-20

Review Team

C. Broadwell

M. Brooks (Author)

B. Glendenning (Chair)

W. Grammer

J. Jackson

W. Koski (Author)

F. Marchet

K. Morita

G.H. Tan

A. van Kesteren

Documents under review:

“ALMA Monitor and Control Bus AMBSI-1 Standard Interface Design Description” Mick Brooks

“ALMA Monitor and Control Bus Serial Interface II Specification” Wayne M. Koski

Minutes

1. General Issues

a) One or two AMBSI versions

Morita introduced the topic by pointing out that several of the questions indicated that the reasons for having two AMBSI were not clear, and there were some worries related to wasted resources in the project. However the replies satisfactorily answered this question. The documents' introductory sections should be expanded to include this information.

Brooks summarized the replies as stating that the desire for AMBSI2 arose for devices that could make good use of a smaller, simpler, interface than AMBSI1. Koski reiterated that minimizing the number of interfaces was still a goal.

Glendenning informed the group that a proposed policy requiring that device designers use one of these

interfaces, with the possibility of waivers being granted for good cause, has been submitted to senior project management for consideration.

? Brooks and Koski to provide introductory material in their documents describing the niche that each design fulfills.

b. Reliability and other testing and analysis

Tan introduced the subject by stating that the main issue is the testing of the circuits, taking into account reliability, EMC, etc. These items will be produced in quite large quantity: ~2000. Fixing them in the field would be awkward. In particular, doing a reliability calculation using some would be very valuable. van Kesteren could do the calculation using software (“Relax”) available at ESO.

van Kesteren outlined the types of information that would be required. This includes components which can be interfaced to databases containing reliability data – the prime information necessary are the component lists. Brooks stated that supplying this, as well as the Orcad drawings, would not be a problem.

Glendenning stated his opinion that while calculations such as these sounded interesting, given that performing them is not yet project policy it did not seem reasonable to make the successful conclusion of the review dependent on them, and instead we should proceed informally until such standards are established. This viewpoint was accepted.

Brooks and Koski stated that they were interested in seeing the results and techniques in producing these results, and perhaps this “test case” would provide useful feedback in formulating the policy.

van Kesteren also mentioned the fact that there will also be EMC and Safety standards for the project that will be required. Tan added that these are topics that he intends to revisit in the near future.

Brooks stated that for AMBSI1 the front ends are the devices most apt to be affected by EMC. The intention is to test the effect of AMBSI1 with the front ends as they will be used in the array (i.e., in metal shielded enclosures). Tan commented that this seems to be acceptable for now. For the final production run we can iterate taking into account Systems standards as necessary.

van Kesteren commented that immunity as well as emission could be an issue. A general procedure is needed (like that which is done for CE testing). There is test equipment at ESO that can be used to test for, e.g., lightning susceptibility. Koski commented that for AMBSI2 the protection will largely come from the device it is mounted to. van Kesteren noted that the CAN bus itself requires attention. [Also see the Galvanic Isolation discussion later.]

? Brooks and Koski to provide van Kesteren with parts lists and other requested information for reliability and other calculations.

c. Bus line grounds

Brooks requested guidance on ground and shield issues for the AMBSI designs. D’Addario in the comments suggested tying the ground to the transmitter. The choice appears to be between not connecting the ground or connecting it to power ground the a transmitter. The solution was not obvious to anyone at the meeting.

Brooks suggested that a jumper be introduced so that both options are possible. D’Addario had suggested that the grounding be through a resistor. Tan did not see the purpose of the resistor. Broadwell also commented that the purpose of the resistor wasn’t clear.

Tan also asked what the AMBSI's are attempting to shield themselves from – magnetic fields? Perhaps a capacitor would be relevant? Koski commented that in the VLBA the monitor and control bus had a shield carried all the way through, but it did not terminate anywhere. Koski also commented that these issues can't be settled separately from the devices themselves which will also have to adhere to standards. For example, Brooks commented that it is still the device designers responsibility to ground the connector shell.

In brief, the situation is not clear and requires investigation.

? Brooks to put jumper in AMBSI1 and investigate these issues further leading to a recommendation for ALMA.

d. Galvanic isolation of bus lines

Marchet indicated that at ESO galvanic isolation is rigorously applied, but after reading the replies he agrees that not providing it in general is correct for the AMBSI/CAN bus, rather providing isolation to particularly troublesome (e.g., any CAN run to the apex) is appropriate. Koski noted that a surge would take out the interface board in any event (e.g., an opto-isolator instead of the CAN transmitter). Brooks commented that this issue is related to the previous discussion on EMC and testing.

van Kesteren commented that opto-isolation would also help cut down on ground loops. Tan stated that as CAN is balanced he is not worried about ground loops.

In the end, we concluded that no action is required.

e. Board cost

Tan introduced the topic by noting that he expects that a \$100 production per cost is feasible for AMBSI1, and less is likely for AMBSI2. He noted that mounting costs are important and surface mounted components should be used and incorporated in the designs when feasible. Brooks asked if this should be done now. Tan responded that it is acceptable to look into this with the last changes before production.

Tan further noted that labor cost can be an issue and it is important to work with specialized companies – without a proper design it could be much higher than a few dollars per board. He further noted that some manufacturers should be able to do testing – at least DC testing, and perhaps more. Grammer commented that board design should include testing feasibility. Jackson commented that these issues need to be considered system wide. Glendenning added that perhaps these issues are most important for the AMBSI's since they may be the largest run of boards.

No short term action is required.

f. Document format

Glendenning stated that as these documents are computing division documents they must conform to the new document template. He will investigate whether some support (e.g., New Mexico Tech student) can be provided to perform the mechanical aspects of the document reformatting.

? Brooks and Koski to format their documents to use the computing division document template.

2. AMBSI1 Issues

a. Number of AMBSI1 connector arrangements/variants

Marchet introduced the topic by asking whether it would be feasible to implement both variants in one PCB. This might help for parts availability and other operational issues. Brooks stated that he is having a technician looking into the possibility of optionally having the 96 pin connector on the same board. Koski commented that the benefit in the field wasn't clear unless both connectors were actually on the board as

otherwise you end up with both variants in the field. Jackson further noted that it might be bad for RFI to have “stuffed” connectors.”

Glendenning asked whether we could eliminate one of the connectors for ALMA. Brooks thought if we kept only one it would have to be the one with the 96 pin connectors as some devices required the extra I/O lines. Grammer stated that various hardware designs would most sensibly use one or the other. Brooks further noted that this could be affected by packaging standards, but those should flow top-down.

The committee decided that nothing should be done to change the current situation.

b. Should the Xilinx controller be programmable by users

Grammer introduced the topic by stating that he considered the Xilinx controller to be a piece of hardware providing critical timing to the interface board. It does not seem desirable to have it under user control. Jackson asked whether the integrity of the AMB could be compromised? Brooks responded that it could – for example it could continuously send resets to devices. He further noted that eliminating this possibility would get four I/O pins back on the 96-pin connector.

? Brooks to remove user programmability from Xilinx controller.

3. AMBSI2

a. Microcontroller selection and use of OTP chip

Broadwell stated that his primary concern was related to the use of OTP (one time programming). He would personally be reluctant to put down OTP because of the possibility of future changes. He would also prefer to not introduce another processor family into the project, e.g. it would be better to use a PIC CAN chip.

Koski agreed with both items. Unfortunately, he was not able to find another chip which would allow for the desired 1”x2” size. The PIC chip has not been released yet, for example. Broadwell asked where the size specification came from? Koski replied that it came from the original AMBSI2 interface designer (Battle), who in turn was reflecting the wishes of Socorro device designers. The small size was considered to be an important design goal. Brooks noted that the fine tuning synthesizer is quite small, for example. Brooks further noted that he doesn’t believe there is an Infineon C167 family with flash memory and CAN on one chip, but this should be checked.

? Brooks to investigate C167 family for chip with flash memory and CAN.

Glendenning asked if a suitable chip is available in the future whether the AMBSI2 could be modified to use it in a “plug-compatible” fashion? Koski replied that it could and should be. It would be plug-compatible on both the device and bus sides.

b. Simplify load modes?

Marchet stated that after reading the replies he does not think the load modes should be simplified. Brooks stated that he feared that state was being introduced into AMBSI2 which has been avoided in AMBSI1. Glendenning asked whether this state could be determined from readbacks from the device. Brooks agreed that it could, but would still prefer that the master did not have to do this.

Koski noted that the AMBSI2 should set its mode by reading information from the device, hence no intervention is required from the master. However the AMBSI2 should also have a standard set of readbacks (e.g., temperature) like AMBSI1.

? Brooks and Koski to determine standard M&C points for AMBSI2

c. Delete device connections other than SPI?

Grammer asked why the AMBSI2 has non-SPI device connections, given that the intent is to configure it over the SPI bus and the assumption that there will be a microprocessor on the other end? Koski stated that it was intended to cater to requests he had gotten from device designers, e.g., to retrieve total power samples which have quite a lot of information to dump requiring a parallel interface. Grammer responded that the AMBSI1 could be used for this application. Glendenning stated that this seemed to be blurring the line between the two designs and thought it would be cleaner to remove the non-SPI connections. Koski agreed.

? Koski to remove non-SPI device connections from AMBSI2

5. Summary and Actions

The panel approves the design of the AMBSI1 and AMBSI2, once the documents are corrected for the accumulated replies and relevant actions, summarized below.

? Brooks and Koski to provide introductory material in their documents describing the niche that each design fulfills.

? Brooks and Koski to provide van Kesteren with parts lists and other requested information for reliability and other calculations.

? Brooks to put jumper in AMBSI1 and investigate these issues further leading to a recommendation for ALMA.

? Brooks and Koski to format their documents to use the computing division document template.

? Brooks to remove user programmability from Xilinx controller.

? Brooks to investigate C167 family for chip with flash memory and CAN.

? Brooks and Koski to determine standard M&C points for AMBSI2.

? Koski to remove non-SPI device connections from AMBSI2

Requirement	AMBSI-1	AMBSI-2
2.1 Data Rates	?	?
2.2 Serial Number	?	?
2.3 Bus Node Addressing	?	?
3.1.1 Latched outputs, sampled inputs	?	?
3.1.2 Parallel interface	?	?
3.1.3 Serial interface	?	?
3.2 Analog Voltage Monitoring	?	
3.3 Device Reset Signal		?
3.4 Device-Specific Features	?	
4.1 Small Size	?	?
3.4 Low Power Consumption	?	?
3.4 Low RFI	?	?
3.4 Low Cost	?	?

Figure 1 - Compliance Matrix for AMBSI1 and AMBSI2

=====
ALMA SOFTWARE DOCUMENTATION REVIEW PROCEDURE
COMMENT LIST
=====

Date : 2001-March-16

Document : ALMA Monitor and Control Bus AMBSI-1 Standard Interface
Design Description
ALMA08001Nx0002A

Author(s): Mick Brooks

Invited reviewers: review preparation

- C. Broadwell (cbroadwell) comment list
B. Glendenning (bglendenning) Chair
W. Grammer (wgrammer) comment list
J. Jackson (jjackson) comment list
F. Marchet (fmarchet) comment list
K. Morita (kmorita) comment list
G.H. Tan (ghtan) comment list

In addition, comments were received from:

- A. van Kesteren (avkesteren)
L. D'Addario (ldaddario)

Review meeting: 2001-March-20 15:30 UT

The following issues have come up from the comment period and should form the basis for an agenda for the review meeting (relevant comment numbers are included in parentheses).

General, pertain to both documents:

- Why are there two standard interfaces? (9, 12, 13)
- What reliability and other testing or analysis should be performed? (6, 7, 8)
- Should these documents use the software document standard? (12)

Specific to AMBSI1:

- Should there be two variants with different connector arrangements. (20, 21, 23)
- Make the Xilinx so that it cannot be user-programmed and is only programmed at manufacture. (42)

---001-----
(avkesteren)

p.0

Is there protection against transients, surges and ESD foreseen ?

REPLY:

Power supply: There is a low dropout regulator on the power supply

input which will provide some protection against transients and surges on the supply. However the ALMA proposed power supply system would outlaw the use of regulators on subsystems such as the AMBSI and there would then be no protection if that proposal is adopted.

Bus lines: There will no opto-isolators on the incoming CAN, reset and 48ms timing event lines. In cases where surge protection is required, isolation will need to be provided external to the AMBSI.

I/O lines: All external bus lines are buffered. I/O lines are not.

ESD: This is more of a procedural issue regarding PCB handling than a circuit design issue.

---002-----

(avkesteren)

p.0

What kind of connectors (including shielding method) will be used ?

REPLY:

Bus connectors are specified in ALMA08001Nx0001 and are either D-sub connectors or circular amphenol connectors. For the AMBSI-1-M, other connectors are pin headers and external device connectors are the responsibility of the device designer. For the AMBSI-1-B, the connector is a 96-pin backplane connector.

---003-----

(avkesteren)

p.0

Considering the PCB design what methods have been used to reduce the risk of EMI (Electromagnetic Interference) ? As an example have the following techniques been considered ?

Tracking of signal lines on the board.

Avoidance of flying leads.

Tracks run orthogonally between adjacent layers.

Use of track mitring (beveling the edges at corners), no track stubs.

All floating conductor areas connected to ground.

Avoidance of slit apertures in ground planes.

REPLY:

The AMBSI-1 uses internal ground and power planes. Track stubs are removed and track mitring is used.

---004-----

(avkesteren)

p.0

What kind of enclosure will these modules have, metallic ?

REPLY:

No enclosures are currently specified. This is the domain of the device designer; in the front end designs metallic enclosures will be used.

---005-----

(avkesteren)

p.0

Has it been investigated (insofar possible) whether components will be readily available on the market in some years ?

REPLY:

The major components are as follows:

PCA82C250: Philips CAN bus transceiver; used in Automotive industry.

C167CR-LM: Infineon micro-controller; used in automotive industry; second sourced from SGS-Thomson

AM29F010B-55: AMD Flash; large volume production but due to small size (1 Mbit) will probably become obsolete in the next few years. AMD upgrade path should suffice.

DS1820: Dallas Semiconductor silicon serial number and temperature sensor; already obsolete; pin-compatible replacement DS18S20 already available and on order.

UPD431008LE-15: NEC SRAM; already obsolete; next size up and new manufacturing technology NEC part is being used for next version.

DS1705: Dallas semiconductor supervisor chip: unknown future availability.

XC9536: Xilinx CPLD; unknown future availability.

The primary concern is the micro-controller; Infineon seems prepared to support this for several years. New versions using this architecture are still appearing and the part is second sourced. RAM and Flash devices remain problematic. If there is any insight on how to choose future-proof devices in this area I would welcome the advice.

---006-----

(avkesteren)

p.0

Will the prototype products be tested on things like vibration, ingress of dust/water, temperature, EMC (Electromagnetic Compatibility), etc. ?

REPLY:

Not formally unless the systems group requires it.

---007-----

(avkesteren)

p.0

Will the products be analysed/tested for safety and is an approval mark (e.g. "CE") foreseen ?

REPLY:

There is no current plan to analyse/test for safety or to gain an approval mark. My feeling is that this is not necessary for the Test Interferometer but should be considered prior to large scale manufacturing for the final array.

---008-----

(avkesteren)

p.0

Has there been any analysis on reliability of the design ?

REPLY:

No. We will comply with any analysis required by the systems group.

---009-----

(fmarchet)

p.0

The document looks well made and rather clear, so I don't have any special

remark.

Many of the comments are just request for additional information especially concerning the microcontroller, since I've never used it.

I have first a general comment concerning both AMBSIs, I and II (the same one has been included in AMBSI-II comments).
>From what I could understand, I see some overlap in the features of the two systems: they both interface with CAN and they both have some capability of performing I/O operations driven by CAN.
One is more aimed at 'high end' applications (no stringent space requirements, flexibility, analog capability) while the other one is aimed at low-end applications (smaller but less flexible).
Although they are similar, they are based on different microcontrollers.

I fear a waste of resources implementing (almost) the same functionality twice and basing on two different microcontrollers, requiring different cultural background, different development environment and so on.
This has also a no negligible impact on production, maintenance, personnel training, spare parts stock, etc.

The only explanation I can figure out for having selected two different environments is that probably some of the designers are more familiar with the Philips devices, while the others are more familiar with the Siemens devices.
Perhaps a relatively small effort from one side could lead to more harmonized designs, making use of the same devices.

I would suggest then, provided this is not judged an excessive effort, at least to implement both cards with the same microcontroller (by the way, I don't have any preference. I don't want to enter the endless struggle whether Philips is better of Siemens).

Even better, I would implement one type of card only, thus simplifying the production, support and maintenance.
If there are still applications requiring a tiny interface between CAN and SPI in which that card cannot fit, I would prefer a completely hardware solution: there are monolithic components (like the MCP2510 from Microchip, an 18-lead SOIC) fully supporting all the CAN protocol features and having an SPI interface.

REPLY:

You are mostly correct in your assessment of the application of the two designs, however the AMBSI-II is intended to be physically smaller, draw less current and to have no external memory devices and thus a lower chip count.

The Infineon micro was selected based on its CAN compatibility and its use in the automotive industry which should guarantee a long manufacturing lifetime. I had no prior familiarity with the device.

I believe that the discussion of the need for two interfcce cards should be explored further at the review meeting. In addition a requirements compliance matrix will be circulated prior to the meeting.

---010-----

(fmarchet)

p.0 General questions:

XILINX FPGA

I understand that the Xilinx FPGA has no specific purpose and can be configured by the user (through the JTAG port) for application specific needs. Is that correct?

Production

Is any test procedure foreseen?

Is any special tool needed for testing the cards after assembling?

Are there (self-)diagnostic features/tools foreseen in the hardware/firmware to help in troubleshooting?

How will the production be managed: in-house assembling and testing, external company assembling but in-house testing, external company assembling and testing?

Will the production be in successive batches, following the needs of the project, or will be in one go?

Is there any estimate of the overall amount of cards to be produced?

REPLY:

The Xilinx has the specific purpose of generating the correct length of CPU reset from the lms Global reset pulse and for generating a CPU interrupt from the 48 ms timing signal if available. If necessary, it can also be configured by the user for application specific needs.

A test procedure already exists and a test jig has been built. We have a workstation configured for testing the circuit board, programming the flash and Xilinx and for testing the standard firmware CAN functionality. There are currently no self-diagnostic features in the hardware or firmware.

For the Test Interferometer, production and testing will be in-house in Tucson. For the production array, the boards will be assembled and tested by an external contractor. Questions about batch production philosophy are premature and should follow an ALMA standard. My opinion is that all boards needed for the array should be assembled and tested in a single batch.

There are 30 AMB nodes per antenna which translates to approximately 2000 boards for the array. The exact numbers of each AMBSI1 and 2 are yet to be determined.

---011-----

(ghtan)

p.0 General

Specifications/performance list of AMBSI-1 is incomplete and should be elaborated so that other users of the AMBSI-1 device can effectively use it. Among others the following is missing:

- Operating temperature range
- Timing of I/O and clock signals, including margins/accuracy
- Rise times of digital signals
- Drive capability of digital outputs
- Load characteristics of inputs
- Signal levels

REPLY:

The operating temperature range will be added. Most of the signal level characteristics are contained in the C167 users manual but some will be repeated in this document for ease of access.

---012-----

(kmorita)
p.0 General

1. Why we need two types of AMBSI.

I think these two AMBSI are complementary to each other. However, it is unclear for me why we cannot cover all CAN-device interfaces with only AMBSI-I. Clear explanation on this point is desirable in somewhere.

2. Design concept

The design concept for each hardware should be described before detailed explanation of hardware design in both documents.

3. Document style

These two documents were written for similar hardwares. But the document styles and explanation scheme are different from each other. At least, the document for AMBSI-II should apply the ALMA document standard.

4. Reference section

There are no reference section in both documents.

REPLY:

1. I agree that an explanation of why there are two AMBSI's but I am unsure of where it should go. We should discuss this further at the review meeting.

2. Additional explanatory text will be added to the two introductions.

3. There is no agreed documentation standard for ALMA. There is an agreed software documentation standard, but it is not clear to me that it should be applied to these hardware documents.

4. A reference section will be added.

---013-----

(ldaddari)

p.0

1. We need to standardize the names and their spelling. I prefer no punctuation and use of arabic numerals: AMBSI1, AMBSI2 (not AMBSI-1 nor AMBSI-II).

2. These specs need to be evaluated in the light of project policies about AMB interfaces, but the necessary policies have not yet been established. To begin with, we need an official response to the review of the AMBSI Requirements document. It needs to be clear whether AMBSI1 and AMBSI2 are to be the *only* interfaces permitted; if not, will there be additional "standard" interfaces and will any non-standard interfaces be permitted? The policy should explain why there are two (or more) AMBSIs and not one, and perhaps should include rules about which one is to be used under given circumstances.

My recommendation is that the two interfaces be sharply and clearly distinguished by device-specific functionality: AMBSI1 is specifically intended to include device-specific functions via custom firmware; its hardware design (processing capacity and I/O capacity) as well as its firmware design (hooks for additional code) are geared toward this. In contrast, device specific functionality is strictly forbidden in AMBSI2. It comes with fixed firmware that is designed to be flexible by including several modes, but it is not customizable beyond the functionality that is built-in.

3. Intro sections of both documents should emphasize what is special about this design, compared with the other one. By reading only the intro sections, an engineer not previously familiar with either design should be able to decide which to select for his application. Specifically, there should be references to the AMBSI Requirements document and an explanation of how this interface meets the requirements; any requirements that are not met should be explicitly pointed out.

REPLY:

1. We will standardize the names to AMBSI1 and AMBSI2
2. The project policy is currently being established.
3. The introduction sections will be expanded.

---014-----
(wgrammer)
p.0
- References to source documents like the Siemens C167 manual and any CAN bus standards document would be very useful.

REPLY:

These will be added including specific references to the bus timing and electrical characteristics.

---015-----
(wgrammer)
p.0
- It's too bad we (NRAO) are once again rolling our own expansion bus interface, when there are some decent well-defined standard buses already out there. It would be nice to use off-the-shelf hardware, particularly for I/O. But I assume that I'm walking on well-plowed ground here.

REPLY:

We are not rolling our own expansion bus, we are using an Infineon microcontroller family bus. The use of off-the-shelf hardware for I/O was never a requirement in the formal requirements document or from my informal discussions with device designers. The CAN bus is the standard we are using for distributing I/O.

---016-----
(wgrammer)
p.3, s.3.1: The signal ground pins should also be identified in Tables 1-7. I'm not crazy about the signal name 'MRST' - looks too much like some kind of reset line. How about 'MRX/STX' (and 'MTX/SRX' for MTSR)?

REPLY:

The ground pins will be added. The MRST is the pin name of the C167 and I believe it is the SPI standard name for this line.

---017-----
(cbroadwell)

P.4 Last line: add "instead of sixteen"
(makes it consistent with the two previous lines)

REPLY:
OK.

---018-----
(fmarchet)

P.4, 1 Introduction
Could the two variants be squeezed in one single PCB by placing both sets of connectors (or better, adding the 96-pin connector to the AMBSI-1-M)? Although this makes the PCB more complex, having one single PCB would simplify the production and the stock.

REPLY:
This is certainly a possibility which will be investigated. Prototypes for both the -M and -B variants current exist and the possibility of merging the PCB will be looked at.

---019-----
(fmarchet)

P.4, 2 General Specifications
In this section, a table clearly stating the specifications (type and number of I/O channels for each variant, amount of memory, speed, environmental etc.) would be useful to summarize the performances of the card.

REPLY:
OK, the compliance matrix and a summary table will be added.

---020-----
(kmorita)

p.4 s2.1
"There are two variants of thw AMBSI-1"

Please explain the reason or the purpose.

REPLY:
I will add an explanation. The reason is that there is currently no packaging standard for ALMA and the device designers who will use the AMBSI have requested both styles of inter-connection.

---021-----
(ldaddari)

p.4 Sec 2.1. By making two "variants" you have really created a third AMBSI. The one with the Eurocard connector ("AMBSI1-B") should be considered a non-standard interface and all mention of it should be deleted from this document. It will not be used in the array. (Several will be used in temporary devices for the TI, so some delta-documentation of those is needed -- showing only how they differ from the *standard* interface. This should be in a separate document.)

REPLY:
A suggestion has been to converge the two connector arrangements to a single layout. Under this scheme it would be possible to optionally install the 96-pin connector on the AMBSI1-M. The layout for this may not actually be feasible on

the layout but will be investigated.

---022-----

(wgrammer)

p.4, s.1, end of 1st par.: A bit unclear. How about "The interface board firmware has by default a minimal set of I/O functions, but can be expanded to use the on-board serial port and devices on the external bus."

REPLY:

OK, suggested wording will be adopted.

---023-----

(wgrammer)

p.4, s.2.1: How about "The first variant, AMBSI-1-M, is designed for embedded use within a bin module. ... The other variant (AMBSI-1-B) incorporates ... instead of multiple pin-header connectors. This variant is designed to plug directly into a standard Eurocard 3U-height subrack, and includes a front panel and associated hardware. ... when compared with the AMBSI-1-M as listed below: ... 12-bit external address bus versus 16-bit addressing"

REPLY:

OK, suggested wording will be adopted.

---024-----

(cbroadwell)

P.5 The two lines connecting the CAN Transceiver and the C167CR box are offset and directions are reversed. Tx is from C167, Rx is into C167

The line from the Low Dropout Regulator to the CPU Supervisor is intended to represent the Vdd line I suppose?

The line from the Oscillator, 5 MHz to XTAL2 of the C167 does not exist, does it?

The Tx and Rx directions for RS-232 are reversed? P3.10 is an output and P3.11 is an input?

REPLY:

Diagram will be fixed for CAN transceiver lines. I will label the Regulator line as Vdd/Vcc. The oscillator line to XTAL2 does not exist; it will be removed from the block diagram. The RS-232 line are also reversed in the diagram and will be corrected.

---025-----

(wgrammer)

p.5, Fig. 1: The D-sub and round DIN connector graphics are misleading. These could be shown as a simple rectangle, with a JPx reference designator matching Fig. 2.

REPLY:

This will be changed. Also a block diagram for the AMBSI-1-B will be added.

---026-----

(fmarchet)

P.6, 2.4 Power

Since the card could be also used as a stand-alone unit, not necessarily in

a crate, perhaps it would be useful to implement some protection against power supply outage (like a fuse and/or a zener).

REPLY:

Noted, but pending ratification of the systems standards for power supplies I would rather wait.

---027-----
(ghtan)
p.6 s2.2
Tolerances on dimensions are missing.
Exact position, including tolerances, of connectors should be added.

REPLY:

OK. Tolerances will be added.

---028-----
(ghtan)
p.6 s2.2
Include dimension drawing showing connector of AMBSI-1-B as well.

REPLY:

OK, extra drawing will be added.

---029-----
(ghtan)
p.6 s2.3
The 96 pin connector is an IEC standards. Please provide the IEC number instead of a manufacturers number.

REPLY:

OK.

---030-----
(ldaddari)
p.6 Sec 2.2, "Mechanical details." This is inadequate; very little mechanical information is provided, much less "details." All we have are two of the outer dimensions (this thing does have some thickness, yes?) and 4 hole locations. Not even the mounting hole diameters are given. The connectors are shown, but their locations are not dimensioned (and the title block says "do not scale drawing!").

REPLY:

Additional mechanical details will be added.

---031-----
(ldaddari)
p.6 Sec 2.3, "Connectors." These are inadequately described. "Standard dual in-line" is not descriptive. You do not even give the gender or the number of pins. Some explanation of why there are 7 separate ones is needed: are they logically separated by function?

REPLY:

Further connector detail and explanatory text will be added.

---032-----
(ldaddari)

p.6 Sec 2.4, "Power." The supply voltage needs to belong to the set of backplane voltages selected for ALMA, and 7-9 V is not likely to be included because COTS supplies mostly skip this range. +5V is much better. The paragraph as written is inconsistent because it says that the required voltage is 7-9 but the current is stated "at 5V."

The concept of converting the board to 5V operation "by not installing the regulator" is not workable. We have to keep a stock of spare boards, and having multiple varieties forces the inventory to be much larger. It is best if the *standard* interface has only one power spec, and all users of it should design to that. However, a small amount of application-specific hardware customization could be allowed if it can be performed by a technician on any board from stock without tools and in very little time -- for example, the installing or removing a jumper.

REPLY:

This whole issue awaits the ratification of a power distribution standard.

---033-----
(wgrammer)

p.6, s.2.2: There really ought to be a figure included for AMBSI-1-B layout as well. Both drawings should be in metric, per ALMA standards, and only need 2 dec. places. It would also be nice to show the Node ID DIP switch location/orientation, the reset switch location, and any LEDs.

REPLY:

OK, drawings will be modified as you suggest and a dimension drawing for the AMBSI-1-B variant will be added.

---034-----
(wgrammer)

p.6, s.2.3: This 96-pin connector has a standard DIN number and type designation; vendor part numbers should *not* be used unless it is a non-standard and/or single-sourced item. (BTW, Panduit is now ITW/Pancon. See the problem?)

REPLY:

I will add the standard DIN number and the IEC number.

---035-----
(wgrammer)

p.6, s.2.4: "..dedicated connector on the AMBSI-1-M,..". Next paragraph; how about "By removing and bypassing the on-board regulator, either AMBSI-1 variants may be optionally powered from a regulated 5V supply."

REPLY:

This wording will be altered: see Chuck's comments.

---036-----
(cbroadwell)

P.7 First sentence: "By not installing the regulator on the board with a single jumper connection..." I am guessing this intends to say something like: "By replacing the regulator on the board with a single jumper connection...?"

Table 1: your note of last Mar said "pins 2, 5 and 9 should be grounded". Is this still the case for pins 5 and 9 (not indicated

as being the case in this table)?

In general, I can't clearly figure out the Global Reset signal. From Table 1, and Figure 1, I suspect it is a differential RS-485 input using pins JP1-2 and JP1-8. Do you then treat one of those as a 5V single ended input to the Xilinx (the one on pin JP1-8)?

REPLY:

Wording regarding regulator replacement by a jumper will be adopted. Yes, pins 5 and 9 will be added to the table. Pin 5 should be grounded but pin 9 should be "No Connection" in case we use that line to supply power over the bus in future.

The Global Reset feature is more clearly specified in ALMA08001Nx0001 and it is indeed a differential RS-485 signal. The next revision of the schematic has a RS-485 receiver to translate the differential signal to a 5V input to the Xilinx. The same is true for the timing signal.

---037-----

(ghtan)

p.7 s2.4

What is the margin on the + 5 V supply for proper operation of the circuit.

Where is this jumper to make the selection?

REPLY:

The voltage margin is dependent on all of the +5V components and will be added. The jumper will be located at the position of the regulator. It needs to be added to the circuit as this option is a recent addition.

---038-----

(ldaddari)

p.7 Table 1: Where is the CAN GROUND signal? It's very important that this be brought to the board, since it provides the bus voltage reference. It should not be tied to power ground on the board, but rather to the line transceiver's low reference.

REPLY:

Ground signals and explanations will be added to the table.

---039-----

(ldaddari)

p.7-8 Tables 1..7: Many of the signals need more explanation. In some cases, like the CAN bus signals, this can be handled by reference to other documents. For many others, a single paragraph of explanation can be applied to a whole group of pins. At least the basic electrical information must be given. (For the analog inputs, you don't even mention the voltage range! We also need the scaling and the coding of the 10 bits.)

REPLY:

Additional information you request will be added.

---040-----

(cbroadwell)

P.8 Table 3 does not show pin 9 for ground on JP3.

Table 4 does not show pins 1 and 2 for JP4.

REPLY:

Ground will be added to Table 3, and Pins 1 and 2 will be added to Table 4.

---041-----

(wgrammer)

p.8, Table 5: Can you say "...7-9 VDC (opt. 5V)" ?

REPLY:

Yes, I can.

---042-----

(wgrammer)

p.8, Table 7: I feel very strongly that the XC9536 programming not be a user option, but that it be programmed before final board test and not changed on the fly by the user in the field (or for that matter in the field at all). I view a CPLD as hardware, only to be altered by the board designer through a documented revision cycle. The I/O bits from the CPLD should be enabled or else deleted.

REPLY:

This is a very good point and should probably be accepted. My aim in including the I/O pins was user flexibility. My proposal is to connect them through the Xilinx as a way of setting location dependent bits and make the Xilinx load a revision controlled part of the AMBSI firmware.

---043-----

(ldaddari)

p.9-10 Sec 3.2: Delete this section; see remarks above.

REPLY:

This section will not be deleted. It was included in order to satisfy a request for a data-sheet style of document.

---044-----

(wgrammer)

p.9-10, Tables 8-10: Please see preceding three comments on Tables 1-7; these apply here as well. On the Xilinx JTAG pins in Table 9: Backplane connector pins should not be wasted on the JTAG port, if reprogramming of the CPLD is not done in the field. These should be left as a pin-header, identical on both board variants. This also frees up 4 extra backplane pins, which if used as additional I/O bits would make both board variants the same for digital I/O (32 bits). Or better yet, they could be used to extend the external bus addressing to 16 bits, as with the -M variant.

REPLY:

Having the JTAG pins on the backplane connector makes the test jig a little simpler but as you point out the pins would not be used in the field. I will remove the JTAG pins and I restore the additional bus address bits.

---045-----

(cbroadwell)

P.11 RCA is referred to here. Should it be defined? The example of writing and reading port P2 at node address 2 also assumes a lot of background information, so I assume the reader is expected to already understand RCA etc.

Table 11, line with RCA 00009-00018: the description could say

"Sixteen 10 bit analog input signals" or something to make it clearer that these are the set of 16 A/D results.

REPLY:

The RCA definition will be added (Relative CAN address). The background information is in ALMA08001Nx0001.
The suggested wording for Table 11 will be adopted.

---046-----

(fmarchet)

P.11, 4 Built-in Firmware

The choice of not using an operating system kernel comes from the simplicity of the software to be implemented or from the unavailability of suitable operating systems?

REPLY:

Real time operating systems are available for the C167. Due to the simplicity of most hardware devices, the use of an O/S was ruled out. The CAN bus and the AMBSI is implementing distributed I/O operations, not distributed computing. I will note the reasons for not using an O/S in the document.

---047-----

(ldaddari)

p.11 Sec 4: What's an "RCA"? (Term is used without definition, and was not used in previous MC documents that I've seen).

What happened to the generic software for I/O to the extended bus? That existed in earlier versions, and I think it should be supported as standard.

REPLY:

An RCA is a Relative CAN address. Definition of the term will be added. The extended bus support is there as RCA range 0x1000 - 0x1FFF. This will be made more explicit.

---048-----

(ldaddari)

p.11 Table 11. The scheme for direction control on the 32 bits of IO is poorly designed, and it is not sufficiently described here. It will create a significant software problem by making the interface have a "state" that must be kept track of. Why is the direction control write-only? How does the direction control work? What is the default if nothing is written to 6 or 7? Which "ports" of the microprocessor are involved is irrelevant here; what matters is which *pins* of the board connectors; you don't even mention that this affects JP7, or which commands affect which pins. See the alternative scheme described in my email to Brooks of early Feb 2001.

The "10 bit analog signals" are inadequately described. How are they encoded in the 2 data bytes of the CAN message? What is the scaling to voltage? What is the correspondence between CAN ID and pins on JP6?

REPLY:

The direction control should be made R/W. More explanation of the direction control will be added.
The analog signal information you suggest will be added.

---049-----

(wgrammer)

p.11, s.4: What does "RCA" stand for? Also, I suggest: "..Further code development will be necessary ... or if the expansion bus is used. The ALMA Computing Group can help..."

REPLY:

Relative CAN Address. This definition will be added to the text. Your suggested wording will be adopted.

---050-----
(wgrammer)

p.11, Table 11: Maybe change "10 bit analog signals" to "Analog input channels 0-15". Also, how is the 10-bit analog input word justified within the 16-bit field? Finally, shouldn't an ability to read the board serial number be included in the generic firmware?

REPLY:

This wording will be changed. See reply to Chuck's comment. Bit justification information will be added. The serial number reading ability is part of the standard firmware as required by the AMB specification.

---051-----
(ldaddari)

p.12 sec 5.1 Library Interface. "API interface" -> "API" (redundant). This description is not at all clear. I happen to know what the idea is because I've worked with it, not because I've read this document. If this style is to be used (and I don't quite agree with it because it gives too much control to the "add on" code and not enough to the "main" code), the description needs to begin by explaining that the writer of device-specific code must actually write the entire application, starting with 'main()', although he will find it convenient to call functions from the library provided. He must compile his 'main()' along with all other device-specific functions he provides, and then he must link them with the library functions to produce executable code. There may be some library functions that he is *required* to call (e.g., 'slave_init()'); if so, this should be stated explicitly, not mentioned as "typically...".

REPLY:

More explanatory text regarding the API will be added.

---052-----
(ldaddari)

p.13 sec 5.2. Suggest changing "user" to "device designer" everywhere. This better describes the engineer's role.

Allowing him to "set the global interrupt enable flag" is dangerous; may he later clear it (Not prohibited by rules in sec 5.4!)? Instead, there should be a library function 'canStart()' or some such thing that he calls when he has finished all his initializations; it enables CAN and global interrupts, and does anything else necessary to begin processing CAN messages. The device designer should be prohibited from ever touching the global interrupt control, or any interrupt registers not dedicated to his device-specific stuff.

The brief list of things to do in 'main()' is not enough. Actually, the structure of 'main()' must be very rigid. Either the device designer should be given model code and allowed to modify it only in well-defined ways, or (better) 'main()' should be fixed in advance and include calls to separately-compiled routines that are device-specific.

REPLY:

user->device designer will be changed.

A "start" routine will be added to the API.

An example main() function is included in Section 5.3.

---053-----

(ldaddari)

p.13-18 example code listing: This is very helpful, and provides lots of insights if the reader will study it carefully. It would be better if there were somewhat more comments and/or some explanatory text.

REPLY:

More explanatory text will be added.

---054-----

(wgrammer)

p.13, s.5.3: "..libraries are available from the ALMA Computing Group." As an additional comment, I think this entire section belongs in an appendix, not in the main body of the specification.

REPLY:

It was initially in an appendix, but previous comments suggested it should be moved in to the document main body. It will now stay in the main body.

---055-----

(fmarchet)

P20, 6.1 Digital Signals

Could the external bus access be configured to multiplex Data/Address in order to save I/O lines (and drivers)?

Is there any mechanism to prevent the (memory) ready signal from blocking indefinitely the CPU (I wonder if there is an exception generating an Illegal External Bus Access trap after a timeout)?

REPLY:

Yes, the external bus lines can be used in multiplexed mode. This was ruled out to make the bus decode logic simpler. Less I/O lines/drivers = more decode logic.

The external bus ready signal can block the CPU indefinitely. The C167 has an internal watchdog system which will be used to detect this fault.

I will add wording to clarify these points.

---056-----

(ldaddari)

p.20 sec 6.1: "... may configure ... for TTL or CMOS" but what is the default? How does the standard firmware configure them? How many loads can they drive?

"... a 4kByte range of addresses is available from standard firmware" but I don't see this in Table 11. Do you mean 4kBytes or 4k words or 4k CAN IDs? How many bytes/words per ID?

REPLY:

Default settings will be added, along with electrical details. "4kBytes" means

4 kBytes of memory and not anything else. The mapping of address locations to CAN messages will be added.

---057-----

(ldaddari)

p.20 sec 6.2: Mentioning "generating analog voltages" in the middle of a discussion of A/D converters is confusing. It need not be mentioned at all; section should be titled "Analog voltage monitoring." There is still no explanation of the encoding of the bits.

The comment "Most analog...will be application specific" is not true; the 16 channels provided here will be adequate for all voltage monitoring within almost all modules. Instead, this paragraph should say, "For special applications where A/D conversion at higher resolution or with precise sampling is needed, or where D/A conversion is needed, it is recommended that the addressed bus I/O on connector JP2 be used to operate external hardware in a memory-mapped manner. Depending on the details of the application, the standard firmware may be adequate to control such devices, or device-specific firmware may be written."

REPLY:

Mention of D/A conversion will be deleted. Suggested wording will be adopted.

---058-----

(wgrammer)

p.20, s.6.2: Should read "Sixteen A/D channels, each with 10 bits..." Strictly speaking there is only a single ADC, with a multiplexed input. Also at bottom of page: An example circuit can and should be provided as an appendix. A possibility would be the analog I/O expansion card just designed and laid out here in Tucson.

REPLY:

OK. I will add these items.

---059-----

(fmarchet)

P21, 7.1 Micro-controller

Is the CPU watchdog used?

What is the actual CPU clock (with a 5MHz quartz several options are possible, if the PLL is used)?

How does the in-circuit debugger work?

Is the development environment provided with an in-circuit emulator or does it use the serial port communicating with the standard microcontroller chip mounted on the card?

How are the Flash memories programmed, by an external programmer or is the in-circuit programming possible?

Reference is made to a tool (Minimon) to be used to program the 'embedded' flash.

My understanding is that this memory is not embedded in the microcontroller (I could not find in the data sheet any reference to it) but rather is the memory on the card. Is my assumption correct?

REPLY:

The C167 CPU watchdog will be used; the DS1705 watchdog will not be used. Actual CPU clock will be 20MHz, internally quadrupled in the C167.

The in-circuit debugger uses the serial port of the CPU to communicate breakpoint, watch and register information to a PC host.
The Keil development environment uses a PC serial port to communicate with the micro mounted on the card.
The Flash memories are programmed in-circuit.
The reference to Minimon relates to the method whereby the external Flash devices are programmed rather than Flash internal to the micro. Your assumption is correct.
I will endeavour to improve the wording of this section to make the points made above more clearly.

---060-----
(fmarchet)

P.21, 7.3 Analog to Digital Converters
Although providing an analog interface including anti-aliasing filters is rather the purpose of an application-specific mating card, putting a simple RC filter at each input line could be enough in simple applications.

The embedded ADC resolution/accuracy is suitable for a large number of applications, but would it make sense foreseen an (optional) external converter with higher performances?
Using the SSC interface, a small 8-pin SMD serial device could be employed.

REPLY:
Analog filtering is explicitly the domain of the device designer. The concept of a standard external higher resolution ADC was dropped due to NRAO's past experience and the lack of any clear requirements consensus from device designers.

---061-----
(fmarchet)

P.21, 7.4 Digital to Analog Converters
The (some of) PWM outputs could be used to implement, together with an RC low pass filter, a simple DAC output.
Alternatively, a small serial converter connected to the SSC interface could be foreseen.
Having one or some DAC outputs would allow closing simple servo loops locally.

REPLY:
Yes. This is in the domain of application-specific uses of the card. I will note in the document that this is possible.

---062-----
(fmarchet)

P.21, 7.7 Connectors
In the -M version there are two similar header connectors (JP1 and JP3). This could lead to confusion or errors: wouldn't it be better use different connectors (as an example a 3 pin strip for the RS232 interface, similar to the one used for the Jtag, but shorter).

REPLY:
Both of these header connectors use the standard 9-pin D ribbon cable to 10 pin header mapping common in PCs. I see no reason to alter this. The connectors will not be disconnected and reconnected very often as the -M version will be inside another module. The possibility of confusion is therefore slight.

---063-----
(kmorita)

p.21 s6.3
What is COTS?

REPLY:
Commercial Off The Shelf. It is a common acronym but I will include an explanation.

---064-----
(kmorita)
p.21 s6.4
Please give us more information or references about SPI and I2C. These interfaces are not familiar to me.

REPLY:
A reference section needs to be added and will include reference documents to these bus standards. They are both serial standards although I2C is intended to be multi-drop, whereas SPI is point-

---065-----
(ldaddari)
p.21 sec 6.3: The use of these ports is not restricted to "COTS instrumentation," so those words should be deleted. It is important that the async port is also used for firmware downloading and for debugging, so designers need to take this into account if they are using it for something else.

REPLY:
Wording will be changed and comments about the use of the async port for code downloading and debugging will be added.

---066-----
(ldaddari)
p.21 sec 6.4: The sync port can NOT be used as an async port; it would be good to make this clear. This should be a 2nd par of sec 6.3 rather than a separate section; unless read very carefully, it looks like there are 2 async ports (sec 6.3) and third port that's synchronous (sec 6.4).

REPLY:
The sync port can be used as an async port given device-specific code. Clarification of the number of ports will be added.

---067-----
(wgrammer)
p.21, s.6.5: I think a GPIB/IEEE-488 port on the AMBSI-1 is not a good idea. It is a specialized instrument bus, which I think would be better handled with separate hardware (preferably some commercial product). I have seen serial-to-GPIB converter boxes, I believe from National Instruments. If any port was to be added in the future, why not something more generic like USB?

REPLY:
I propose to say in this section that GPIB will not be supported by the AMBSI-1. There is no requirement to do so.

---068-----
(wgrammer)
p.21, s.7.3: How fast can the converter sample?

REPLY:

I will add the fact that the conversion time is 10 usecs.

---069-----

(ghtan)

p.22 s7.9

Estimated cost is for single of quantities. Try to make an estimate for an optimized design (e.g. only surface mount components) when it is produced in large quantities using automated assembly equipment.

REPLY:

All components which make sense are surface mount already. The difference between quantities of 1-10 or 2000 are negligible. If we use automated assembly we will have to pay for testing also and I suspect that the final per-board cost would be similar. Nevertheless, I will add this estimate.

---070-----

(ldaddari)

p.22 sec 6.5: GPIB. Omit this completely. There is no need to say what is not included, unless it's something called for in the Requirements document, which this is not.

REPLY:

This section will be omitted.

---071-----

(ldaddari)

p.22 ff sec 7: Implementation Details. I did not have time to read this section.

Missing:

-- Provision for external node ID determination, as specified in Requirements Document. This is important. (Sec 7.8 mentions expanding DIP switch setting from 6b to 11b, but it's necessary to bring some bits off board.)

-- Any support for the timing signal. Apparently it connects to JP1 but then goes nowhere. There needs to be a provision for causing a timing event to generate an interrupt for a device-specific interrupt handler. The standard firmware might include a handler that merely increments a counter whose value can be read as a monitor word; this would be useful.

REPLY:

There are 4 I/O bits available for external node ID determination. This will be added to the document.

Support for the timing signal will be added to the document. It is in the next hardware revision and just needs to be updated here.

---072-----

(wgrammer)

p.22, s.7.5: There is no mention of the output current source/sinking capability of the digital I/O bits. This ought to be somewhere in the document. Also, I take strong issue with omission of detailed bus timing and loading specifications for the expansion bus. This is absolutely essential for proper design of I/O expansion cards that use this bus. It

can be included as an appendix or even referenced as a separate document.

REPLY:

I will add the I/O current handling specifications. The bus specifications are in the C167 User Manual. The reference will be added.

---073-----

(wgrammer)

p.22, s.7.6: Does "bus master" refer to the CAN bus? This is a little unclear to me.

REPLY:

The bus master in this context refers to the ABM (VxWorks) computer which is reading monitor data and sending control data. I will try to clarify the wording.

---074-----

(wgrammer)

p..22, s.7.7: This is redundant; all of this was previously covered in section 2.

REPLY:

Agreed, it will be removed.

---075-----

(wgrammer)

p.22, s.7.8: Suggested change in last line: "This can be expanded to 11 bits in future revisions."

REPLY:

OK.

---076-----

(wgrammer)

p.22, s.7.9: I would suggest specifying the board unit cost for production quantities (100-500), which would include the cost of the board plus parts and necessary supplies and tooling for assembly (i.e., solder mask, paste). The labor should not only include board assembly but also final burn-in and test time; this estimate should also include an approximate additional cost per board if assembly is contracted out. I would also cost each board variant seperately; my guess is that AMBSI-1-B is more expensive due to the DIN connector and panel hardware.

REPLY:

I will try and improve the cost estimates for production amounts and include the additional cost components you suggest.

=====
COMMENTS PERTAINING TO SCHEMATICS ONLY

---000-----

(cbroadwell)

P.0

General comment: on P.4, I notice C12 specified as 0.01uF instead of 10nF, while many (but not all) 0.1uF caps are specified as 100nF. I prefer uF, since my brain is not used to nF, but perhaps this is an American point of view?

REPLY:

I believe that using whole nF rather than fractional uF is the ISO preferred standard. It is common practice in the US and Australia to avoid the use of nF. Since nF is a standard unit, the references should be made internally consistent anyway.

---001-----
(cbroadwell)

P.1 I/O pins on C167 and CAN blocks are wrong directions
(related to P4.5 and P4.6 being reversed)

REPLY:

Yes, this has been fixed for the next revision.

---002-----
(cbroadwell)

P.2 P4.5 and P4.6 are reversed;

You said you were dubious about the opto isolators, are they still in the circuit?

REPLY:

The next revision has the pin reversal fixed and the opto isolators have been removed.

---003-----
(cbroadwell)

P.3 Pinouts on the 10 pin CAN connector are wrong.

REPLY:

This has also been fixed for the next revision.

---004-----
(cbroadwell)

P.4 Pin 99 should be tied to ground.

P4.5 and P4.6 module ports directions reversed.

You said you probably would put pullups on all 32 digital I/O lines (Ports 2, 7 and 8)?

I am sure there is no problem, but on Pin 84 (Vpp), the C167 data sheet says it is NC, so I left mine floating, while you connected yours to Vcc.

REPLY:

Pin 99 (EA) is grounded in the new revision and pullup resistors have been added to the I/O lines. Connecting pin 84 to Vcc appears not to cause any problems.

---005-----
(cbroadwell)

P.8 R18 and R19 values have the letter "R" at the end. Is this intentional?

REPLY:

No. It will be fixed.

---006-----

(cbroadwell)

P.10 You said P3.0 has a "4k7" (4.7K I assume) pull-up?

REPLY:

Yes, added to next version is a 4.7 k ohm pull-up resistor on P3.0.

---007-----

(fmarchet)

p.11 Schematics

On sheet 11 of the schematics the Chip Select input of U14 is labeled as P6.1 negate, while the module port is tagged P6.1.

REPLY:

The negation will be removed. P6.1 is a port designator; the signal is used

---008-----

(cbroadwell)

P.12 You said Pin 8 should not be connected to pins 1 and 5.

R24 value ends in "R".

REPLY:

Yes the use of the DS1705 watchdog has been removed in the next revision. Resistor labels will be made more consistent.

_____oOo_____

AMBSI-2 Replies

Chuck Broadwell

>Comments for Wayne, on AMBSI-II memo

>General comment: need page numbers printed out

OK

>General question: I glanced at the microcontroller and it appears to have 32KBytes
> of One Time Programmable EPROM. Do you intend to develop software
> in external memory and eventually "commit" to code to burn into
> chips before surface mounting? I assume this is the case for now.

Correct. But, I'd love a Flash version of this.

>P.1 Section 1: I did not know what a Philips PXAC37KBA was; perhaps add
> something like "micro controller" or whatever?

OK

> Section 2, bullet 6: I do not understand this; what does it mean to
> "handle 32 Control or Monitor messages directly"? Why are the messages
> specified to be eight bytes, instead of 1 to 8 bytes?

The Philips device can predefine 32 distinct CAN messages via an internal DMA/CAN controller. This might not be that important now. But a thumbnail sketch of the device can't hurt where CAN is concerned.

The eight bytes was due to my preception that all messsages would be exactly eight bytes long. This will be changed to one to eight bytes.

> Section 2, bullet 7: "Going directly to the" not a
> complete sentence?

This was clarifying the previous sentence. Suggestions?

> Section 2, bullet 7: "Power can be applied at either **the**
> 10 pin SIP or the 40 pin DIP connector." (Add word "the"?)

Yes.

>P.3 Section 3, bullet 2: add "micro controller" after PXAC37KBA here also?

Yes.

> Section 3, bullet 6: "For J2 the pin definitions could **be** changed.."
> (Add the word "be".)

Yes.

> Final paragraph, first sentence: "The operation of the AMBSI-II",
> sounds funny to me. I'm not exactly sure what it means to say that the
> operation consists of the interface etc. (Sorry to be dense.)

Well, I like it. Would eliminating "operation of the" read more smoothly?

> Third sentence: the word "converts" should be singular (convert)?

Hmmm. I think that converts fits due to the fact that two signals (Reset and 20.833Hz) are being converted. However, it might be wise for me to separate them. Do you agree?

>P.6 Why not assign J1 and J3 pin numbers identically, i.e. if J3
> pin assignments are specific for easy mapping to a DB-9, why
> does J1 have different pin assignments (or why does it exist
> at all except that it also has 5V power on it)?

Early implementation problem. Prior to this version, the AMBSI-II had only two connectors J1 and J2. J3 was added by Mick's suggestion. J4 was driven by Larry D'Addario. And J5 was added due to an oversight by me. Because people here were using J1, I couldn't eliminate it.

>P.9 Two places: change "will consist:" to "will consist of:" ??

OK

> This will be general: There is discussion of "given that
> each byte is transferred in a period of 2.5 us, this would
> allow about 8.3 us between the end of one transfer and the
> start of the next". I do not know where the 2.5 us comes
> from, but assume it is the time for 8 bits at 4 MHz clock
> rate, plus some overhead of 0.5 usec. I also do not know
> where the 8.3 usec comes from. $8.3\text{ us} + 2.5\text{ us} = 10.8\text{ usec}$,
> but this confuses me. I can guess the 10.8 usec is perhaps
> supposed to be related to the bit rate on the CAN bus and is
> intended to be the transfer time of one byte over the CAN
> serial bus? If this is the case, then it sounds as if you
> expect to receive bytes one at a time from CAN and forward
> them, one at a time, over SPI (meaning you expect to receive
> one byte from CAN and forward it to SPI while the next byte
> is coming in from CAN)? Most likely I am way off base
> here, so perhaps I just am missing some details that are
> keeping me from getting in gear. But if my description of
> "forwarding one byte at a time" is what you intend, I'm a bit
> confused by it. In the next paragraph, I attempt to describe
> my understanding of how you will receive bytes from the CAN
> bus:

First, the byte period is 2uS, instead of 2.5uS. My mistake. For monitor requests, I'm allowed 150uS from the end of the CAN request to the start of my response. So I must communicate the request to the device, and receive up to eight bytes within that period. The 8.3uS is the time between each byte on the SPI bus. This allows the other microprocessor time to load/store the byte, without the need to handshake. Then, I can formulate the CAN message to the computer. I'd like to see the 150uS go to 200uS, but I know that may not be allowed. I could run commands slower over the SPI, but I'd like to maintain a single set of timing on the SPI.

I've been working with the SPI protocol and some changes will occur.

> In case there is any relevance, I would expect the following

> approximate timings: A CAN message consisting of a full 8
> byte payload (128 data frame bits + any stuffed bits) will
> take ~ 130 usec minimum transfer time over the CAN bus. A
> CAN message consisting of a single 1 byte payload (64 data
> frame bits + any stuffed bits) will take ~ 66 usec transfer
> time. The message payload would not be available for
> transfer to the SPI bus until all payload bytes had been
> received and the message object is declared complete. In my
> preliminary tests, the fastest I could transfer an 8 byte
> payload was closer to ~155 usec and a 1 byte payload was ~
> 100 usec. (If the worst case padding were required, there
> would be one pad bit for every five data frame bits, thus a
> 128 bit frame theoretically could become ~ 150 bits or
> more.)

> So, my fundamental confusion is where did the 10.8 usec come
> from? What is pacing the AMBSI-II to User Device SPI
> transfers at the 10.8 usec rate?

As before, the 10.8uS was the time to transmit a byte and padding to
allow the device time to load/store the byte so that handshaking is
unnecessary.

>General thoughts:

>Since I have been working with the Infineon C167 processor myself, and
>have only been exposed to the Phillips controller based AMBSI-II for
>the first time today, I am trying to figure out for myself the reasons
>for implementing it.

>The first thing that strikes me is the apparent fact that the AMBSI-II
>will use a micro controller that does not have Flash for program code
>(so there is no In Circuit Programming capability). I feel that this
>makes the development of the program code difficult, since you must
>reach a point where you are ready to really commit a particular
>version of the code to production.

>Obviously, the parts count for the proposed AMBSI-II is smaller, and
>board space is smaller, than a corresponding system based on the C167.
>But the fact that much software development has already been done for
>the AMBSI-I, and the fact that Flash memory is used, and In Circuit
>Programming can be done, makes me wonder if there are sufficient
>applications for the Phillips based AMBSI-II to justify the completely
>new design and use of a different processor and software development
>environment.

>The Phytex MiniModule 167 (a C167 module with 256 KBytes of Flash and
>256 KBytes of SRAM) is a 6.9 square inch board compared to the
>AMBSI-II board area of 2.5 square inches, or a factor of nearly 3
>times as much board space. I believe a C167 based AMBSI-II board
>could be designed in an area no larger than the Phytex module, and
>perhaps slightly smaller. Still, the Phytex, at 3.3 inches by 2.1
>inches is fairly small.

>A C167 based AMBSI-II module could still use SPI, Async Serial, or
>parallel interfaces to the target processor or hardware.

>The production price of a C167 based version would certainly be

>higher, and I have no idea what quantities we are looking at, so
>perhaps there is a cost basis for the Phillips approach. But my
>present feeling is that unless the justification is primarily
>production cost, then perhaps a C167 based version makes more sense
>from a design/development perspective. The C167 may be overkill in
>many applications. But again, it must come down to a trade off of
>design/development costs versus production costs.

This is the question that we need answered. Why two?

The other aspect that popped up here was, why any? Why not, build
the device controller and the CAN controller together using just
one processor?

The Computing Group, rightly so, didn't want a dozen different
approaches on how communications would be on the CAN bus. So,
they wanted standardized approaches.

Then, the call went out for a simple CAN to device interface.
Compact, simple interface, and a PCB board drop in. John
Battle at ALMA-NM promoted the SPI style of interface and
so was AMBSI-II conceived.

When John Battle left, I was requested to work on this. The
main driver for me was a 1 by 2 inch board to do this in.
And I was able to come very close to that.

Mick Brooks may be able to provide some more insight into
this situation as he probably saw more of how it happened.

So the review is to answer AMBSI-I only? Allow AMBSI-II?
merge AMBSI-I and AMBSI-II into one solution? Something
entirely new?

Chuck:

>A)
>>> Section 2, bullet 7: "Going directly to the" not a
>>> complete sentence?

>> This was clarifying the previous sentence. Suggestions?

>Not really, since looking at it further, I get confused. I believe the
>reference to "One 10 pin SIP or the 10 pin DIP is used for the CAN
>interface lines" must be talking about J1 (the SIP) and J3 (the DIP)?

>It appears that these provide two options for connecting to the CAN
>bus. J3 pinout maps to a DB-9 connector for connection to the CAN bus.
>J1 is another connector for connecting to the CAN bus (?).

>So the statement: "Going directly to the nine pin D-Connector via
>ribbon cable, or via the ten pin SIP to the user device." is where I get
>confused again. If this section is talking about J1 and J3, the two
>connectors I understand to offer a choice of connector pinout to the CAN
>bus, why does the statement refer to one going to the CAN bus and one
>going to the user device? (I thought they both went to the CAN bus, you
>just have a choice of which one to use.)

To simplify, J1, J2, and J4 are meant to connect to the User Device's PCB. J3 and J5 are connections on the AMBSI-II component side. Therefore for J1, the CAN Bus would connect onto the User Device's PCB somewhere, and then be routed to J1. You can bypass the User Device's PCB by going direct to J3 instead. This is what I'm trying to indicate. The AMBSI-II was meant to be a drop-in board and the main connections to it are J1 and J2.

>(Regarding J1 and J3, in my later question about pinouts being >different, I did not even realize one was a SIP and one was a DIP, since >on the schematic they both appear to be SIPs to me.)

I will change the schematic.

>B)

>> Final paragraph, first sentence: "The operation of the AMBSI-II", >> sounds funny to me. I'm not exactly sure what it means to say that the >> operation consists of the interface etc. (Sorry to be dense.)

> Well, I like it. Would eliminating "operation of the" read more smoothly?

>How about:

>"The AMBSI-II connects to the CAN Bus at either J1 or J3. The >electrical interface is provided by the PCA82C250 CAN Bus Transceiver. >This chip provides....."

OK

>C)

>> Third sentence: the word "converts" should be singular (convert)?

> Hmmm. I think that converts fits due to the fact that two signals

> (Reset and 20.833Hz) are being converted. However, it might be wise

> for me to separate them. Do you agree?

>I also went for a second opinion on this, and Jim pointed out what I had >missed:

>"Also at J1 or J3 the MAX3082's..." <----- this is the singular >possesive.

>I believe you want the plural (2 chips), non-possesive, which is how I >read it.

>So, I believe you are saying: "Also at J1 or J3, the MAX3082 chips >convert the two RS-485 differential input signals (Reset and 20.833Hz >Timing) to TTL."

>I still believe convert should be singular: two chips--> convert, one >chip--> converts.

You're correct. I had one chip previously instead of two, which is how that wording came to be. I'll change this too.

>D)

> The other aspect that popped up here was, why any? Why not, build
> the device controller and the CAN controller together using just
> one processor?

>In our correlator control cards, we chose the C167 in part because it
>was already selected by Mick. Ray would have much preferred to stick
>with his old, familiar 87C51 type chip. But I felt that we had
>sufficient requirements that a "more powerful" processor might be nice,
>and that it would be a reasonable goal to attempt to standardize across
>the project. Ray still feels that the 87C51 could easily do everything
>we needed. (I have been very pleased with the C167 and the Keil
>software development tools. The C167 plus one Flash memory and one SRAM
>provide a wealth of resources.)

Well, you better not tell Ray that the chip I'm using is an improved 87C51 based chip by Philips.

>I also felt we should attempt to use CAN in the correlator, again for
>uniformity. But since our processors have other high priority tasks,
>rather than being dedicated only to CAN as the main task, there is the
>concern of conflict.

I understand.

>I have designed in an alternate SPI based multi-drop RS-485 bus for
>possible use, if we run into problems with CAN going to each of our
>control cards, but so far I doubt that we will need it.

Interesting.

>In the tests I have done so far, I have felt the need to depart from
>Mick's CAN routines somewhat. I believe it is necessary for me to set
>the CAN interrupt priority higher than any other, to insure that we do
>not lose messages. But it is also necessary that I strictly limit the
>time in the CAN interrupt routine. We will have 16 msec interrupts that
>will be critical. So I will place my 16 msec handler at one level lower
>priority than CAN, but will restrict how much work can be done in the
>CAN interrupt handler. I believe this means that I cannot use Mick's CAN
>code directly, since I believe his "call back functions" all operate at
>interrupt level.

>I also have implemented "correlator specific" protocols for broadcasting
>messages to multiple nodes. In many cases, we will have up to 16 nodes
>that all require the same identical commands, so this is another area
>where I need to depart from the standard M&C code. But as much as
>possible, I am using Mick's code and modifying it only where needed.

>My main test so far has been to transfer blocks of data to a CAN node,
>in 8 byte messages. When I send the messages as fast as possible from a
>C program in one C167, the interval of time to transmit one message is

>approximately 155 usec. The length of time I spend in the receiving
>interrupt handler, to copy the 8 bytes to a buffer, is 45 usec. Thus
>when multiple messages occur rapidly, it costs me 30% of the available
>time to merely receive and stuff the data into a buffer. I believe this
>will be a manageable situation. (This all used C code, no assembly. If
>need be, I will attempt to decrease the 45 usec by using assembly.)

>So, it is still my intention to use the one processor for time critical
>correlator control tasks and for all CAN control and monitor
>operations. But it is my understanding that in the correlator we are
>not required to precisely implement the M&C bus protocols, and this
>provides me the flexibility I may need to accomplish this.

>In your case, I understand that there was the thought that the user
>control processor might not be able to handle the CAN interface along
>with other time critical operations. It is not clear to me that
>converting the CAN messages into SPI messages actually un-burdens the
>user processor very much. It seems the same number of messages still
>arrive at the same rate and require the same response.

Yes, and there was this isolation question. As I understand this,
the Computing Group wants to isolate or buffer the User Device away
from the CAN Bus. And the desire is to avoid specialized CAN Bus
implementations. In fact what you have to do for the correlator is
breaking the rules as far as I know. I know full well the complex
needs of a correlator, which is why you'll be allowed to do a best
fit, but I doubt that any more exceptions to the rules will be made.

>Also, I understand that the "user processor" is likely to be a Microchip
>PIC. I wonder why the PIC MCP2510 was not chosen for the CAN function,
>since it appears to implement a "CAN to SPI" interface. (I have not
>looked at the chip myself, but on the surface, it sounds like the same
>thing as the AMBSI-II.) (I have used PIC chips at home for some years
>now, and really like the newer flash devices. I seem to recall that
>John was perhaps using something like the 16C74 at one time, and would
>hope he migrated to the newer 16F87X family of ISP flash devices. It
>sure is handy to just hook a cable to a board, throw a switch and
>download new code.)

I looked at the MCP2510. That would have been easy, if not for the
isolation concern. The MCP2510 is meant to interact with a
microprocessor as a SPI slave. Therefore User Device would have to
interact with it via interrupts and the SPI Bus when a CAN Message
occurs. Thus the User Device would be an active partner in dealing
with the CAN Bus.

Microchip is developing a CAN Flash device, so perhaps in the future
an improved AMBSI-II could exist.

>E)

>>General question: I glanced at the microcontroller and it appears to have 32KBytes
>>of One Time Programmable EPROM. Do you intend to develop software
>>in external memory and eventually "commit" to code to burn into
>>chips before surface mounting? I assume this is the case for now.

>> Correct. But, I'd love a Flash version of this.

>My gut feeling is that I would not want to design in a OTP surface mount
>device!

Yes, but if it is limited, then for the time being, this would suffice for the test array. That is, a small interface that can be plugged into User Device PCB's.

>Well, I suppose I have bent your ear more than enough, so, good luck!

I'm always willing to listen, and respond.

Fabio Biancat Marchet

Fabio:

>Comments on "ALMA Monitor and Control Bus - Serial Interface II

>Specification, issue Jan 17 2001.

>The document looks clear so I don't have many comments, only, perhaps, a few
>words could be spent to describe the firmware.

>I have first a general comment concerning both AMBSIs, I and II (the same
>one has been included in AMBSI-I comments).

>From what I could understand I see some overlap in the features of the two
>systems: they both interface with CAN and they both have some capability of
>performing I/O operations driven by CAN.

>One is more aimed at 'high end' applications (no stringent space
>requirements, flexibility, analog capability...) while the other one is aimed
>at low-end applications (smaller but less flexible).

>Although they are similar, they are based on different microcontrollers.

Correct. Also the need to have a small as possible interface.
Initial size spec was 1 x 2 inches.

>I fear a waste of resources implementing (almost) the same functionality
>twice and basing on two different microcontrollers, requiring different
>cultural background, different development environment and so on.

>This has also a non-negligible impact on production, maintenance, personnel
>training, spare parts stock, etc.

True, but as the AMBSI-II is so simple, I don't see that much impact.

>The only explanation I can figure out for having selected two different
>environments is that probably some of the designers are more familiar with
>the Philips devices, while the others are more familiar with the Siemens
>devices.

>Perhaps a relatively small effort from one side could lead to more
>harmonized designs, making use of the same devices.

Not really. I couldn't see a method to implement the Siemens device
in such a small size.

>I would suggest then at least to implement both cards with the same
>microcontroller (by the way, I don't have any preference...I don't want to
>enter the endless struggle whether Philips is better of Siemens...).

First, I don't see redoing the AMBSI-I with the Philips chip. As

above, I don't see a way to implement the AMBSI-II with the Siemens. We don't have a microprocessor war, rather how to best carry out each interface as we understood the requirements to be. If a AMBSI-II can be designed with the Siemens that fits the current size/footprint, then I'm for it.

>Even better, I would implement one type of card only, thus simplifying the >production, support and maintenance.
>If there are still applications requiring a tiny interface between CAN and >SPI in which that card cannot fit, I would prefer a completely hardware >solution: there are monolithic components (like the MCP2510 from Microchip, >an 18-lead SOIC) fully supporting all the CAN protocol features and having >an SPI interface.

The main reason for the AMBSI-II over MCP2510, is the isolation aspect. As I understand this, it is to isolate the user from the CAN as much as possible. Or to limit the User interactions with the CAN bus in a known and defined way using authorized interfaces. The MPC2510 interacts directly with the CAN Bus and User device. Depending upon the User Device programs needs versus anothers, this could cause differences in responses to the CAN Bus. The Computing Group is concerned about this issue. If they were not, then the MPC2510 would have been used long ago, and the AMBSI-II wouldn't have been conceived.

>P.5 diagram
>No galvanic insulation is provided for the CAN lines. I think it would be >very useful to solve any possible ground loop problem.

Refine this if possible.

>P.5 diagram
>How is the CAN address configured (is it 'frozen' in the OTP)?
>Some of the I/O pins could be connected to dip-switches or jumpers for >flexible address configuration (although they should share some functions).

The AMBSI-II will aquire the CAN Address from the device over the SPI Bus using a specialized request. By assigning the CAN Address to the device and then sending to the AMBSI-II allows flexibility. If you swap out an AMBSI-II with another, you don't have to look up the address to set it. If you have two identical devices in an antenna, you can swap them and the proper CAN Address would be loaded for each unit dependant upon where they are plugged in at. Without opening the module up. This method has worked very well for VLBA providing technicians with easy methods to solve problems.

P.5 diagram
>The reset signal (RST/) generated by the MAX705 supervisor is an input for >the User system.
>Is there any case in which the User could require a reset of the AMBSI-II?

The RST/ signal goes to the AMBSI-II's microprocessor and to the User Device. It isn't an input, rather it is an output. I put this in, in case the User System might like to have a free power on RESET via a supervisor. The only problem is that if the global reset is used then both the User System and AMBSI-II would be reset. But many felt that this was a minor problem.

>P.5 diagram

>The watchdog integrated in the MAX705 supervisor could be used to increase
>the reliability of the system (provided one I/O line of the microcontroller
>can be used for this purpose).

The microprocessor I use has a built in Watchdog.

>P.8, 5. User I/O Features

>On the market are available CAN to PC interfaces, they include also the
>software for the PC.

I don't understand this point.

>P.14 AMBSI-II Usage

>It is not clear to me what 'load a mode' (or sub-mode) means. Is that just a
>selection of existing (i.e. firmware) routines, or is it a real firmware
>download?

>I would say that must be the first case, since there is not possibility, as
>far as I can understand, to download any firmware.

>I'm not sure I understood correctly.

This isn't a firmware download. Rather, an User Device to AMBSI-II
setup. For example, this would load the CAN Address, Address Size,
Fixed or Variable Length Monitors, etc. Taking Address size, this would
tell the AMBSI-II if it need to send a single byte, two bytes, or
three bytes of address to the User Device for proper operation.
This is done in order to provide flexibility for the User Device.

>General questions

>Firmware

>No mention on the firmware architecture, how to generate it, how to download it.
>Although this is mostly a hardware description, I would put some basic
>information concerning the firmware and the tools to be used to develop it.

I can provide this, if it is considered necessary.

>Production

>Is any test procedure foreseen?

>Is any special tool needed for testing the cards after assembling?

Similar to VLBA, a test stand would be devised to fully test the
AMBSI-II.

>Are there (self-)diagnostic features/tools foreseen in the hardware/firmware
>to help in troubleshooting?

I would like to see features in the AMBSI-II that would read back
CAN Bus status as well as User Device interaction status. Also
the DS18S20 temperature section as well as the serial number. Finally
the AMBSI-II has a RESET/ (not RST/) function that has to be dealt
with. So I do see diagnostic issues for both. How to implement
them is the question. I have some strong views on this, which may
take some face to face communications to convince others.

>How will the production be managed: in-house assembling and testing,
>external company assembling but in-house testing, external company

>assembling and testing?

I prefer in-house for all. Better communications, quality control, etc.

>Will the production be in successive batches, following the needs of the >project, or will be in one go?

I prefer one monstrous go, but expect batches.

>Is there any estimate of the overall amount of cards to be produced?

Not for AMBSI-II as of yet. However approximately 5/antenna might be a reasonable guess for now.

-wayne-

Arno van Kesteren

- > * Is there protection against transients, surges and ESD foreseen ?
- > * What kind of connectors (including shielding method) will be used ?
- > * Considering the PCB design what methods have been used to reduce
- > the risk of EMI (Electromagnetic Interference) ? As an example have
- > the following techniques been considered ?
- > o Tracking of signal lines on the board.
- > o Avoidance of flying leads.
- > o Tracks run orthogonally between adjacent layers.
- > o Use of track mitring (beveling the edges at corners), no
- > track stubs.
- > o All floating conductor areas connected to ground.
- > o Avoidance of slit apertures in ground planes.
- > * What kind of enclosure will these modules have, metallic ?
- > * Has it been investigated (insofar possible) whether components will
- > be readily available on the market in some years ?
- > * Will the prototype products be tested on things like vibration,
- > ingress of dust/water, temperature, EMC (Electromagnetic
- > Compatibility), etc. ?
- > * Will the products be analysed/tested for safety and is an approval
- > mark (e.g. "CE") foreseen ?
- > * Has there been any analysis on reliability of the design ?

The AMBSI-II is mainly meant to be a drop-in board so some of the above has to be covered by efforts in module design.

I've done the PCB using many of the techniques as above. As much as possible, but still remain flexible and to carry out the basic functionality.

Components have short life spans now compared to the past, so that remains a problem in all aspects of ALMA.

Testing is an aspect that would be done for AMBSI-II. The CE is a question for the systems group. I don't foresee that though.

Reliability analysis? Generally I've not seen this at NRAO. All I can say is that I build carefully, and I have responsibility

for equipment that has been in operation since 1976 for VLA and since 1986 for VLBA. I build for longevity.

Gie Han Tan

>Many thanks for preparing the AMBSI-2 documents. I would like to submit
>the following list of comments on these documents:

>AMBSI-2 Specification:

>p.1 s2

>Power supply is +5 V, but what is the allowed margin on this for proper
>operation?

The general tolerance +/-200mV. However, it needs to be specified exactly.

>p.2

>Please indicate tolerances on the dimensions shown.

Yes, plus I need to bring my docs under ALMA standards if I continue.

>p.3 s3

>Clock oscillator frequency is proposed to be 29.4912 MHz for standard
>baud rates. Baud rates for which interface, the serial I guess?

The serial is correct. I don't expect a need for serial, but I desire to test it out.

>General

>Specifications/performance list of AMBSI-2 is incomplete and should be
>elaborated so that other users of the AMBSI-2 device can effectively use
>it. Among others the following is missing:

>-Operating temperature range

>-Timing of I/O and clock signals, including margins/accuracy

>-Rise times of digital signals

>-Drive capability of digital outputs

>-Load characteristics of inputs

>-Signal levels

I agree. If AMBSI-II continues, then this would be done.

>General

>Optimize the circuit design so that only surface mount components can be
>used. This will give a great saving in time and costs when a large
>volume of the device is produced using automated equipment.

The bulk of the design uses surface mount save for the headers which connects to the User Device.

>General

>Try to make a cost estimate of the manufacturing costs, including
>materials, for large volume production.

Cost of material would be around \$100.00. Too soon to guess on manufacturing.

Wes Grammer

>All: Need page numbers and numbered figures.

I agree.

>At end: Would be useful to included a References section for pertinent
>supporting documents, like the PXAC37KBA data book, SPI bus spec., CAN bus
>spec, etc.

For every item utilized, I will include the PDF for it. I prefer to detail what I do.

>s.1: The premise of your design approach comes across as weak and overly
>vague. Are you saying a single-processor approach (like Mick's) lacks
>robustness? Is the purpose here then to "buffer" the CAN port from the
>device controller with well-tested generic hardware and firmware, in case
>it dies? How likely is this, and how does your design approach address
>this potential problem? Your statements in this section are qualified and
>seem to lack conviction. You need to make a more persuasive case here.

No. Mick's design is excellant. I too have a single processor approach, so I don't understand exactly what you mean by robustness. The AMBSI-II is a low end interface, whereas AMBSI-I is a high end interface. The AMBSI-II is meant to place onto the CAN Bus a known quantity. The same I believe can be said for AMBSI-I. The fear is that if the User processor is also linked to the CAN Bus directly, then CAN communications from device to device might differ too greatly and cause disruptions. The idea that different devices might have different software/interrupts and thus those differences might be translated onto the CAN Bus.

>s.2: The mechanical details of your design deserve a small subsection of
>their own, covering details of the interface connectors, board footprint,
>etc. On the mechanical drawing: 1) It should be in metric, per ALMA
>standards; 2) Interface connectors should be clearly identified by J#; 3)
>A side view (showing req'd min. height) is needed; 4) An actual size
>(while impressive) is not needed; and 5) The size of the mounting holes
>(or recommended *metric* screw size) should be specified.

I agree, if AMBSI-II is allowed to continue, then the documentation must also continue and to be brought under ALMA documentation specs.

>s.2: DIP and SIP usually refer to device packages, not connector types. I
>would suggest calling these "dual-row header" and "single-row header"
>intially, and simply "header" collectively afterwards (I assume you are
>using header pin arrays).

You can see the terminology of DIP/SIP for headers too. Not a big deal for me. I'll defer this to the systems group.

>s.2: There needs to be a general system-level description at the top of
>the section, on how the AMBSI-II interacts with a device controller and
>the host (CAN) controller. You can include a very brief and concise
>description of the SPI bus, possibly cite the original bus specification
>document if it exists, and mention industry support for it.

If I can locate this, I'll be glad to put it into the documentation package.

>s.2: You mention implementation details that are covered in Section 3.
>Keep your electrical description more general here. You also should
>specify the environment (operating temp range).

I agree that this will be refined.

>s.3, #2: Should identify this as a microcontroller either here or in s.1

Yes.

>s.3, #6: Most of this should already have been covered in s.2.

No. Section 2 describes what is there. Section 3 is meant to describe what it does. I'll try to improve this.

>s.3, #7: How about "standard serial baud..."

Yes.

>s.3, All: The structure of this entire section is very convoluted and hard
>to follow. Why not bread it down into subsections, each covering one block
>within your design and how it functions in detail? It would also make it
>easier to reference from another document later on.

As above.

>s.4: It is redundant to specify the # of pins in each connector here.
>Again, I would give each connector a subsection#. Alternatively,
>converting your textual lists to numbered tables with contents put into
>columns, or using actual graphical layouts of the connectors with the pins
>numbered and labeled is much easier to reference and understand. Details
>like the Reset Duration can be footnoted. More detailed description really
>belongs in the other sections.

I like to list out connector functions. As a former technician this proved to be invaluable for tracking down I/O connection problems. The graphical idea is a good one. Redundancy isn't always a bad thing.

>s.4: Does "Single-Pin I/O" refer to an alternate (programmable) function
>for the pin as a bidirectional I/O bit? If so, this needs to be made much
>more clear (it is only briefly mentioned in s.5).

Correct. But beyond the SPI aspect, I have not fleshed out the other aspects.

>s.4, J2 description: Could the User Device also be a peripheral chip with
>an SPI bus, as well as a microprocessor (i.e., and A/D converter)? This
>sentence really belongs at the front of s.2, in the general system-level
>description.

It is meant to be a microprocessor. SPI peripherals would need special attention.

>s.5: The opening paragraph is way too fuzzy. What is precisely meant
>by "overburden"? A designer using your board needs better-defined
>limits to work with, if this really is a concern.

Fuzzy in Fuzzy out. The whole concept of multiple interface techniques isn't clear to me. I simply did all I could to meet the AMBSI requirements documentation. That document set forth all the design goals, but not the how/why of it. I can see many advantages for this, but not much on what exactly should be done. So I'm making my best guess. The AMBSI-II simply started out as a CAN to SPI interface. The other ideas grew out as possibilities, which became requirements. The AMBSI-II is very flexible now, but the implementation in certain areas remain fuzzy. After all, we need time to explore and prototype with real hardware in order to determine what works and toss what doesn't.

>s.5: This whole section is really in the wrong place - it needs to be way >at the beginning, right after your introduction. These highlight the main >features of your board, and belong first.

>s.5, #2: It should be made very clear that external hardware is necessary >to support RS-232, RS-422, or RS-485 standards.

"The interface circuitry is a burden upon the User Device for any specialized circuitry." What more needs to be said? I'll be glad to be redundant as needed.

>s.5, #4: How is this accomplished? You need to elaborate at least a >little bit how the user configures the hardware for this configuration, >since these pins seem to have multiple functions. Also, what type of I/O >(CMOS or TTL) it is, and available drive capability.

As above in my Fuzzy rant. Also interface circuitry is a burden... Configuration is totally unknown, save for SPI.

>s.6A, #4: If the SPI bus permits multiple slave devices, you may want to >require an open-collector/drain buffer at the slave end.

A SPI Slave has this built in.

>s.6B: Stuff about RS-232, etc. is redundant if included in earlier >section, and is irrelevant to timing.

Maybe, but if this isn't tossed, then the baud rate is pertinent for proper operation and is a timing consideration. Also provides me a path to describe start, bit length, parity, and stop bits.

>s.6, All: Bus timing should include setup, hold, and max. delay times. >Output drive capability and max. recommended capacitive loading or bus >length should be specified.

If AMBSI-II is continued, then this will also.

>General comments:

>Overall this document seems more like a report on development of a >specific piece of hardware. It just doesn't read like a well-defined >interface specification. I think the overall premise (as I understand it, >a simple, robust and generic CAN interface) needs more development and >thought.

Development and thought. I almost lost it with that. I was requested to do this activity in addition to my other duties with VLBA and VLA. I have done my utmost to carry out this task with plenty of development and thought. Yes, it does read like a report. It does need refinement. But, the big question is should AMBSI-II be part of ALMA? This document should provide everyone enough information to make that decision. If it moves forward, then this document will be expounded upon in great detail. Hardware can be tested and problems found, solved, or bypassed. If it doesn't go forward then I'm probably done.

>While the hardware details of the AMBSI-II are highly developed in this >documents, very little if any mention is made of the software side. For >example, what kind of software development system is available for >modifying or adding code, and whether it is even the responsibility of the >user to do code development for the board. This needs to be clarified.

Little is still known about the software side. My plan is assembly language programming. As I program, I work with the Users and we can both solve problems on both sides.

Larry D'Addario
AMBSI2 Comments:

>p.1 sec 1, Intro: Very inadequate. See common comments, above.

This depends upon whether the documentation is separate or not. If these are merged by some means, then a different section would cover the differences and why to use one over the other. If kept separate then a designers guide should be developed to aid in that and placed in the appendix for both.

The AMBSI1 and AMBSI2 document/section should describe each device specifically.

>p.1 sec 2 item 7, connectors: "SIP" and "DIP" are not proper designations for >these connectors. They refer instead to IC packages. The figure on p2 seems >to show only 4 connectors, not 5, and they don't all match the pin counts >given here.

I've seen this terminology for connectors, but this can be changed. Wes Grammer had a good thought for this.

>Why is the subjunctive used in several places, but the indicative in others? >"RS-485 Reset is used" but the board "would ... handle 32 messages" and the ID >"would be ... a DS1820 or equivalent." It "would" if what?

Because I don't know what is required in many cases. The AMBSI requirements documentation forced me to produce a very flexible device, but didn't define the means for me to implement that flexibility. So where I knew exactly what I had, I tried to indicate that, and where I had to guess, I indicated that too. I guess I'm too

truthful. Looking at the DS1820 issue, it was a different device when I started, changed to the DS1820, and now again changed to a DS18S20 because the DS1820 designation is obsoleted by the company. So you'll have to excuse me for being wishy-washy when everything else is too.

>p.3 sec 3, list: Here there are also lots of subjunctives, saying what >"could" or "might" be done. The whole point of this device should be that its >features are fixed and well defined. Although sec 2 said that this device >"has been designed" it now seems that much of its design is left to the >future. Not even its functionality is well known, much less its >implementation.

As above. Per the AMBSI requirements I have a high degree of flexibility, but no clue as to how to implement that. So I take my best guess, with the hope that someone who wishes to utilize this device can help me define limits or determine if those features may indeed be of use. Many aspects of the AMBSI will be driven by the needs of the device designers as well as the Computing Group. So stating that this device is fixed and well defined doesn't fit with the requirements that allow a high degree of flexibility. I mean the AMBSI2 started as a simple CAN to SPI device. Then the requirements changed it to a device that could do SPI, parallel, serial, and direct I/O. If we left it as SPI, then I would have been more decisive because I wouldn't have so many possibilities.

>p.6, J1: Which pin is CAN ground? Preferably, it should NOT be tied to the >board ground plane (power ground) except perhaps through a resistor.

Perhaps through a resistor? What if I don't connect at all? And if I don't connect, why define a pin?

I guess Mick will think this through and provide an answer to me.

>p.6, J3: What is the purpose of these redundant connections?

Trying to be nice to the Device Designer. I don't have to solder anything here, if it is not required.

>p.8, sec 5: Again, we have a list of "some of the possibilities" rather than >a description of how the device actually operates (will operate?). There are >numerous critical design issues that are not addressed. For example, if >various modes really are supported in the one fixed device, how is the current >mode selected?

Right now the mode is passed into me via the SPI connection. I make a special request and the Device designer tells me what I need to know.

>p.9 sec 6B, Serial Bus: This is too vague to be of any use. Is an >asynchronous serial port supported by the firmware or not? If so, exactly >what are its characteristics?

Yep, another possibility. If this is a bad idea, then this mode can be tossed. However, this might lend itself to a CAN TAP, where I TAP into the CAN Bus and send information (oneway) to a laptop, PC to look at CAN Bus activity. I think this may be an interesting idea for troubleshooting or even a maintenance aid. But I don't know, I just think about this stuff.

>p.14 sec 8, Usage. The concept of controlling the mode of the interface via
>CAN control messages is seriously flawed. It means that not even the
>interface, which ought to be simple (much less the rest of a complex device)
>can be relied upon to behave predictably after a power cycle, reset, or when
>being driven by inadequately debugged software. At the very least, each
>interface must have only one mode. If the AMBSI2 is capable of several modes
>(a small number only), the actual mode should be fixed upon installation.
>This can be done with removable jumpers, or by a small number of the device-
>side pins.

First cut at development. Now the SPI via that special request is how
I intend to work this solution. I hate jumpers. Last resort for me.
Prefer Plug and Play to jumpers, dip switches, etc.

>Missing:

>-- Any discussion of how node address is determined; method for having
>the node address determined externally.

>-- Any mapping of CAN IDs to interface functionality.

Didn't know how to implement this at that time. Hated the dip switch
idea. So via that special SPI request, the Device will send me the
CAN ID. Makes for a Plug and Play situation for both the device
and AMBSI2.