NATIONAL RADIO ASTRONOMY OBSERVATORY

Charlottesville, Virginia

October 7, 1974

VLA COMPUTER MEMORANDUM #115

CPU LINK SORTWARE
Specification

Barry Clark

To make the Modcomp synchronous system work as a unit, it is imperative that one CPU be capable of addressing a device attached to another, sending data by the parallel link between CPU's. This memo describes the software (device handler and associated task) which supports this capability.

I. External Appearance.

The logical device is named LK. To support the logical device the resident task LKT must be present and active in the two systems. Device handlers and tasks LKT in the two systems are identical.

To assign a logical unit to a device on a remote CPU, the logical unit is assigned to the link. Then, within the remote CPU, the logical unit must be assigned to the appropriate device within the task LKT. For example, you are executing a background program from Boss, and wish to assign SI to the card reader on Monty. Your job control deck would include the statement

```
$ASSIGN SI=LK
```

Sometime before this was executed, you would have pressed console interupt on Monty, and, on his console device, typed,

```
/LKT/ASSIGN SI=CR
```

II. Line Discipline Conventions

The hardware link makes available five communication bits, set by select commands, in addition to the 16 bit parallel data lines. Select bit 11, status bit 5, indicates remote device error, and will be reserved for that purpose in the software. The remaining four bits will be used to maintain the line discipline. Three of these, select and status bits 13, 14, and 15 will be used to request an operation,

Regarding them as a three bit number these lines have the following significance:

2	Receive I/O request
3	Transmit I/O request
4	Receive I/O acknowledgement
5	Transmit I/O acknowledgement

An acknowledgement is an I/O request generated by the task LKT to request that the completion of the requested I/O be posted. The remaining bit, select bit 10, status bit 12 has significance only if the originating link data set is busy. Then 0 = initialized in register transfer mode, 1 = initialized in DMP mode.

Service interrupt is generated not only on device termination, but also when the select bits of the other computer are changed. The internal SI is generated by a terminate command, either generated by the handler routine (data transmition complete, ready to terminate current operation and schedule next operation) or by the I/O system (start first operation). Line discipline is maintained by requiring different responses to internal SI in the master and subordinate CPU (specified by a bit in the physical device table).

On receipt of an internal SI, the master looks to see if there is another node (that is, I/O request), and if so, immediately selects status 3 or 5 as appropriate, and initiates transfer.

A transmission requires at least three passes through the service interupt routines. In the first pass, in the originating CPU, the request bits (described above) are set into the select lines. This triggers the service interupt in the responding CPU. In the first pass in the responding CPU the acknowledgement is placed on the lines (i.e., a '3' is acknowledged by placing a '2' on the line). This change causes a service interupt in the originating CPU which performs pass 2. In pass 2, the originator checks to see if a valid acknowledgement was received, and, if so, Enters transfer initiate output with the select bits 13-15 reset to zero, and with bit 10 set. If there was not a valid acknowledgement (both CPU's think of themselves as originators), The responding CPU checks the state of the master/slave bit (bit 15 of word 15 of the PDT). If master, he serenely waits for an acknowledgement, if slave, he changes his status to acknowleged the master's request. Pass 2 in the receiver causes a transfer initiate, input and connection of the data interupts.

III. Information Sent and Received

When an I/O request is initiated and begins service, eight words are sent describing the request. They are:

0	The REX code
1	The I/O count (if data is to be transmitted)
2	I/O Node Address in sending computer
3-7	First 5 words of the UFT

This information is sent to a buffer area in the opposite I/O handler. If the REX was a write, this information is followed by the DMP trnasmission of the buffer data to a buffer area in the opposite I/O handler. Detect bit 10 is reset before the DMP transfer is initiated. This is used by the transmitting computer to make sure the receiving one is initiated before doing his own transfer initiate. On the completion of this operation a service interupt is generated. The I/O node is removed from the PDQ and the device handler in the remote system marks the data present. The task LKT, examining these marks, then executes the REX command contained in the request, using the UFT sent from the other system. On the completion of I/O, the UFT busy flag is reset, which causes LKT to enqueue a write request, which will be interpreted by the handler as a request to generate an acknowledgement.

At the start of an acknowledgement, the same eight words of control information are returned to the originating CPU (into a buffer in the handler). If the REX was a read, this information is followed by the DMP transmission of the read data into the requesting task's buffer. The task's UFT is then updated with the file status, position index, and transfer count sent from the remote CPU. The node is purged by the usual process of restoring it to the available queue in the calling task. All buffers associated with this I/O request are deleted.

IV. Caveats

No error checking is done on the accuracy of transmission over the link.

The I/O node linking is not preserved by the task LKT. Therefore, if two I/O requests, with two UFT's, are simultaneously addressing the same device, the order in which the data is output to the device may not be the order in which the I/O requests were generated.

Response to error conditions is not yet specified (failure of line discipline, insufficient buffers) and at present may cause halts and aborts.

My intention is to extend the handler to enable core-to-core transmission. This is not yet specified.

The maximum length record which may be sent by link is 128 words.