Title

HIGH SPEED SERIAL LINKS FROM A MODCOMP COMPUTER SYSTEM TO A TEKTRONIX 4041 GRAPHICS DISPLAY TERMINAL

Authors

Meng, John
Gin, Fong.

Publication Date

1980-05-01

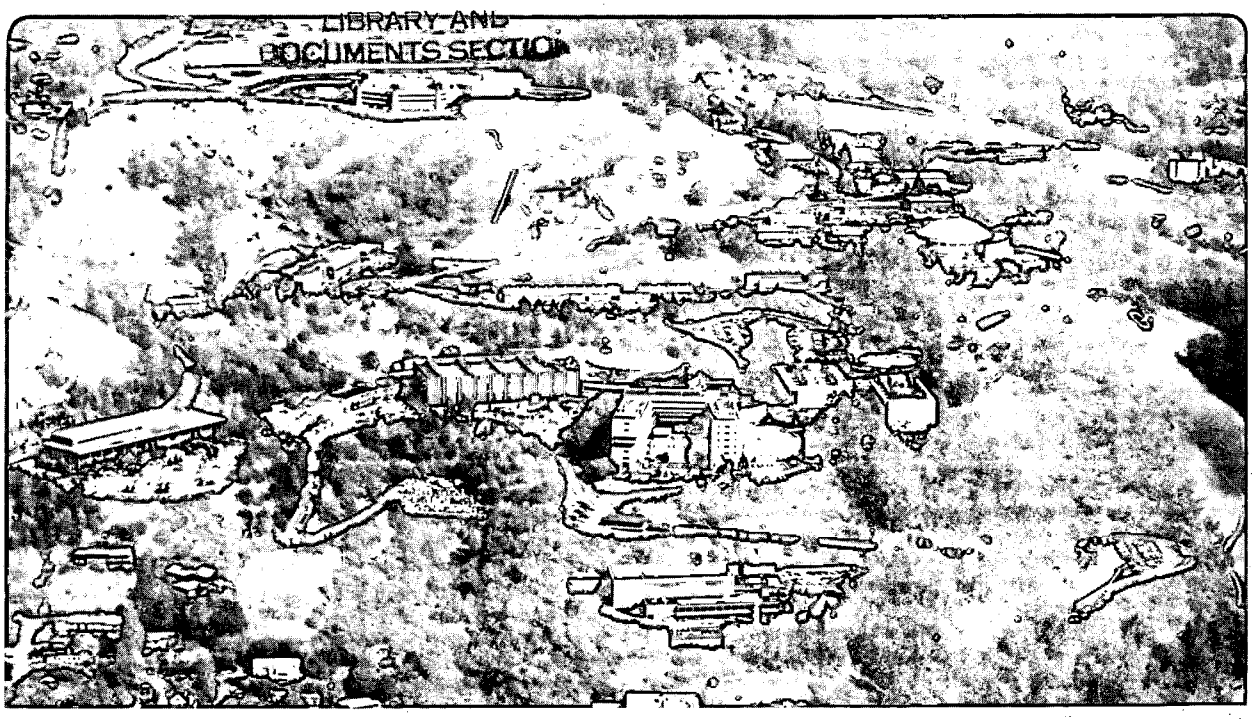
Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

Engineering & Technical Services Division

HIGH SPEED SERIAL LINKS FROM A MODCOMP COMPUTER SYSTEM TO A TEKTRONIX 4041 GRAPHICS DISPLAY TERMINAL

John Meng and Fong Gin

May 1980



Prepared for the U.S. Department of Energy under Contract W-7405-ENG-48

DISCLAIMER

HIGH SPEED SERIAL LINKS FROM A MODCOMP COMPUTER SYSTEM TO A TEKTRONIX 4041 GRAPHICS DISPLAY TERMINAL*

John Meng and Fong Gin
Lawrence Berkeley Laboratory
University of California
Berkeley, CA 94720

Introduction

Two high-speed computer-to-terminal connections have been implemented. One (21X153) is implemented via modifications and additions to the ModComp 4811 controller. The other (21X258) is an independent DMP-driven controller. Both controllers funnel into the same data line to the 4014 terminal, and subsequently avoiding interference between the two is a function of the driving software. At the 4014 terminal, the Tektronix data communication's interface card is modified to accept the higher transfer rate (21X1532-H-3A). Part of the modification is the addition of logic to effect handshaking between the terminal and the driving controller.

The data link between the two devices is asynchronous. At the terminal, the asynochronous receiver is driven from the 4.9 MHz terminal clock, divided by 16. At the computer, the asynchronous transfer is driven from the ModComp 5 MHz input/output clock, divided by 16. The two are crystal controlled and close enough in frequency to provide for reliable asynchronous data transmission and reception.

Since transmission from the terminal to the computer is primarily keyboard driven, it was not considered worthwhile to increase the transmission speed in this direction. The input to the 4811 controller from the terminal is therefore unmodified, and the DMP controller is not bi-directional. It only outputs data.

* This work was supported by the Physical Research Division of The Department of Energy under Contract No. W-7405-ENG-48.

Changes made to the 4811 do not affect the operation of ModComp diagnostics for the controller. They affect only port B. Port A is untouched.

Changes at the 4014 (21X1532-H-3A)

Handshaking circuitry has been added to the Tektronix-supplied data communication interface card, the handshaking signal being supplied to the REQ TO SEND line at the cable connector from M14, a 7437 buffer. A low at this point prevents data transmission from the computer. Note that REQ TO SEND is a TTL signal capable of driving a 100 ohm terminated line and is not a ±15V signal.

The OR gate (M13) forces REQ TO SEND low if the terminal is busy ($\overline{\text{TBUSY}}$ false) or if the terminal is in the midst of strobing data from its shift register ($\overline{\text{TSTROBE}}$ false) or if the start bit of a transmission has reached the end of the shift register (M13, pins 4, 5) and the terminal is not set to $\overline{\text{LOCAL}}$ (LOCAL true).

NOTE

Placing the terminal in LOCAL will always force REQ TO SEND true (high). Data transmission to the terminal is not stopped, but the data is ignored.

***

Two clock circuit modifications have been made. First, the 4.9 MHz terminal clock is inverted and wired to the start-bit detector in place of the 614 KHz clock. The inverter is one section of the 7437 (M14, pins 4,5 and 6). Second, a pin has been added to the row of baud-rate selection terminals (labeled EXT) and connects to the gated 4.9 MHz clock. Gating has been added to the clock to guarantee that a character just received will not be immediately overwritten by a succeeding one. If system handshaking is function properly, this gate is not necessary.

Changes to the 4811 (21X1542-H-1A1)

Changes to the 4811 controller simply replace the transmit part of the UART with a TTL implementation capable of generating the required higher data transmission speeds. Four pins on the UART chip (22,23,24,25) have intentionally been bent to prevent their making contact with the socket. The signal on pin 23 (XU3K-23) would normally strobe data into the UART. It is connected instead to the input of an inverter and to the set input of a flip-flop (center, left-side of 21X1532-H-1A1). The output of the inverter strobes the data (low true) into a holding register (2-SN7495's). The flip-flop starts a transmit sequence. It's $\overline{Q}$ output is the OUTPUT BUF EMPTY signal to the 4811 which goes false. This same signal releases the RESET signal on the ÷16 counter via an OR gate and 936 DTL inverter. The ÷16 counter clock is driven from the 5 MHz input/output clock appearing on the 4811 (XU5P-10). The output of this counter drives one input of a 3-input gate. Another input to the gate comes from the Q output of the flip-flop, being enabled when the character is accepted for transmission. The third input to this gate is true when the controller is ready to transmit a character (DATDON- is false and TERM NOT BUSY is true). Note, also, that the hardware echo must be disabled (XU4R-5 TRUE; XU4R-6 FALSE) for data transmission. When the three-input gate produces an output pulse, the inverted data from the input buffer is strobed (through the 8267B multiplexer) into the output shift register consisting of three SN7495's.

Ten bits of this shift register (8 data+START+STOP) are monitored by the combination of a 7430 8-input gate and a 7411 3-input gate. When the shift register has completed a transmission, the gate inputs are all high, producing a low level at the output of the 7430 (DATDON-).

When the shift register holds a character to be transmitted, DATDON- is high, enabling the connection of the clock (5 MHz ÷ 16) to the shift pins (SHIFT) on the 7495 shift registers through the 7408 gate near the bottom center of the drawing. The resulting serial data stream passes through the 7408 at the top right corner of the drawing, is inverted by the 7401 output gate and passed to the high-voltage (±15 V) line driver through an OR gate in the DMP interface (21X258 H-7). The 7408 gate mentioned above AND's the DATDON- signal with the data to produce the required output level between transmissions. The 7401 gate is used to prevent hangups if hardware echos are attempted simultaneously with DMP transfers.

Hardware-generated echos can be enabled by program and are implemented by special connections detailed on 21X1532 H-1A1. Specifically, the ECHO ENA being true (ECHO DIS false):

(1) disables the -16 counter whenever the output shift register is empty (7401 gate at lower left corner of drawing);

(2) allows the generation of a load pulse to the output shift register when the UART signals the successful reception of an input (on XU3K-19);

(3) and selects (via the 8267B multiplexers) incoming data from the UART to be the source of data for output.

Generation of the load pulse (item 2 above) occurs near the center of the drawing. The BDTRDY signal from XU3K-19 sets a flip-flop which then produces the rising edges of the strobe signal (through the NAND and NOR gates). This high level passes through three 936 DTL inverters (used for delay) and resets the flip-flop to produce the trailing edges of the strobe signal. The strobe loads data into the output shift register, starting a normal character transmit sequence.

Drawing 21X1532 H-2 details the cabling between the terminal and the controller, the miscellaneous wiring changes required in the controller, and handshaking logic.

DMP Controller 21X258

Logic for the DMP output connection to the 4014 terminal is contained on the logic card added to the opposite half of the ModComp frame containing the 4811 controller. With exceptions of the final ORing of data outputs, the DMP controller's tapping of 4811 I/O bus connections, and a cross connection to prevent interference, the 4811 and the DMP controller are completely independent. Buffers for I/O bus signals required to drive the DMP controller are detailed on 21X258-H3.

Drawing 21X258 H-1 contains:

(1) logic used to decode the device number (generates SELN, low true);

(2) logic to decode CPU I/O commands (OCMDN, ISTAN, ODATN; output command, input status, output data; all low true);

(3) logic used to produce the necessary control signals: INTEN- interrupt enable: EOBLK - end of block: TERM - terminate: XFINI - transfer initiate;

(4) Logic to detect this device having priority (TUHPRI) and to drive succeeding levels of priority (DTDO8N in the lower right corner of the drawing);

(5) and logic to generate the source identification code when an interrupt is requested (IDON is pulled low to generate code 20_{16}).

The controller priority level is hard wired to 9. Changing this requires using more or fewer sequential DFB lines to generate PRIO9N and also requires changing the DTD line number driven from the 7407 in the lower right corner of the drawing.

Device address 20_{16} is decoded by the 9301 chips at the left of the drawing, and may be changed if desired by using different pins on these decoders to drive the 7402 NOR gate. The 7427 3-input NAND is used to decode device zero for the benefit of the DMP processor, and its inputs should not be changed. The interrupt vector may be changed from 20_{16} by changing the ID line drive shown at the top right corner of 21X258 H-1. One additional device number is associated with the DMP controller. This device number tells the DMP processor which set of word count/memory address registers to use. The DMP terminal controller is hardwired to DMP device 5 via the ID3N - ID4N - ID5N connections on drawing 21X258 H-5.

Drawing 21X258 H-2 contains interrupt generation logic both for service interrupts (SI) and for data interrupts (DI). Each interrupt channel contains an enable flip-flop, set to the enabled state by the combination of INTEN and a data bus bit (the right-hand flip-flops) on the drawing), and a request flip-flop (center of the drawing).

The service interrupt request flip-flop is set if enabled and a terminate occurs (TERM or LOCTMR). The one-shot also fires at the end of the last transmission from a buffer because DMPX goes low at the end-of-block signal and BUFMT subsequently rises when the last character transmission is completed. Data interrupts are requested if enabled, when the DMPX signal falls at the end-of-block.

Interrupt requests (SIRN and DIRN) are sent to the computer when the request flip-flops are set. The computer returns an update signal (SIU, DIU and their inverse-- SIUN and DIUN) which tests the priority chain via SIREQ or DIREQ driving DTD08 on H-1. PRIO9N signals this device having priority and enables the reset of the request flip-flop at the end of the inverted update pulse (DIUN or SIUN). On H-1, upper right corner,

SIREQ or DIREQ anded with TUHPRI provide the interrupt identification code (20_{16}) on IDON.

Drawing 21X258 H-4 details the generation of status bits which are read by the input status command (ISTAN). Bit 8 (DTDO8) is set while the controller is processing data. (DMPX is set by the transfer initiate and reset by the end-of-block). Bit 7 (DTDO7N) is set by HOLDC (H-7) being false. HOLDC is false when the tramsmit buffer is not empty and the DMP controller is active (instead of the 4811). Bit 7 changes with every word transmission whereas bit 8 only changes at the beginning and end of a block tramsmission.

Bit 1 (DTDO1N) is set if transmission is interrupted before a zero word count. Bit 4 (DTDO4N) is set if the DMP processor detects a memory parity error while reading data from computer memory. Bit 0 (DTDOON) is normally set unless an error condition (bit 4 or bit 1) occurs.

Drawings 21X258 H-5 and 21X258 H-6 together comprise the controller's DMP control section. Drawing H-6 details the generation of reset signals which are mainly used on drawing H-5.

The CPU normally generates a transfer initiate command (XFINI) to begin operations. The logic on H-5 uses this to initiate the DMP processor and to automatically begin a sequence of data-transfer requests (DMPRQ). The sequence of data-transfer requests is stopped by the arrival from the DMP processor of an end-of-block signal (EOBLK) which causes the controller to send a terminate request (FINI) back to the DMP processor. Referring to the central region of drawing H-5, the upper pair of flip-flops controls transfer initiate requests, the middle pair of flip-flops controls terminate requests and the bottom pair of flip-

flop controls data transfer requests. Each flip-flop pair functions in similar fashion. The first is set by an incoming request. The second is set from the first by a synchronized edge derived from the combination of the DMP request flip-flop setting and an I/O clock edge (SYNDMR goes high). The first flip-flop of the pair is then reset by the AND of the output of the second flip-flop and the high level of the synchronizing flip-flop-- SYNDMR (see H-6: signals RESTN, RESDN, RESIN). The OR of the three left flip-flops (H-5) is used to set the DMP request flip-flop at the bottom left of the drawing (DMREQ).

Transfer initiate, data requests and the terminate all generate DMP requests. The intent of the request is sent to the DMP processor via the state of the source ID lines, ID1N and ID2N. The transfer initate request is initated by program (XFINI sets the left-hand flip-flop) and data transfers are initiated by either the end of a transfer initiate cycle (INSETN) or by the end of a data word shipment to the terminal (DMPRQ, originating on H-6 when the data output shift register is empty). The terminate request to the DMP processor is initiated by an end-of-block pulse (EOBLK) whose source is normally the DMP processor by way of decoding on H-1.

The flip-flop at the bottom right on H-5 is set whenever this controller actually has control of the I/O bus for its DMP operation. The ENZN signal from this flip-flop feeds back to 21X258 H-1, device decoding, and enables the use of device code 0 by this controller. When the DMP processor wishes to communicate with any device, it does so by using device code 0. In this way the DMP processor generates input data, output data and output command (for terminate and end-of-block) signals to an active controller.

Drawing 21X258 H-7 contains the output shift register for data being sent to the display terminal. Logic near the top of the page controls the shift clock (IOCLK -16) via handshaking with the terminal and via buffer-empty signals (BUFMTN and HAFMTN). The chain of inverters at the bottom of the drawing accepts the output data command (ODATN) from H-1 and uses the DTL inverter delays to produce the necessary pulse edge sequence to parallel load the shift registers both with data from the data bus DFB0 - DFB15) and with start and stop bits.

The data bus is 16-bits wide and each word contains two ASCII characters. The most significant half of the word (DFB00-DFB07) is sent to the terminal first, followed by DFB08 - DFB15. Since a DMP transfer is one word minimum, transfers to the terminal are two ASCII characters minimum.

Accurate control of the shift clock is achieved via the clear connection to the counter (HOLN). The counter divides IOCLK by 16 to produce SHCLOK. Releasing HOLN allows production of the clock. The clock must be stopped under any of three conditions:

(1) If the DMP controller is not in control of transfers to the terminal (DMPCN is true) and the character output buffer is empty (BUFMT is true); or

(2) If the terminal is busy (TERB is true) and the buffer is half empty (HAFMT is true, meaning one of the two characters has been sent); or

(3) The character transmit buffer is empty (BUFMTN is low).

Case (1) produces HOLDC, (2) produces HOLDA, and (3) produces HOLDB, and the OR of these passes through two additional gates to hold the

counter reset. The first of these two gates prevents clock resetting in the midst of a clock output pulse (SHCLOK) thereby quaranteeing each clock pulse to be full width. The second gate prevents clock stoppage if the start bit is on the output line (DATLI set).

HOLDB is driven from a flip-flop, and is released when the terminal is not busy (TERNBN high) and the buffer contains a character or two (BUFMTN high). The case of HOLDA presents special problems because the buffer always starts with two characters for transmission. At the half-way point, the second character is instantly ready for transmission. To allow more time for TERB to stop the transmit clock, delay is placed in the connection from HAFMT to the hold-release gate for HOLDA.

SOFTWARE

The enclosed CRT symbiont was written for a ModComp IV running MAX IV. As such it may not run verbatim on other ModComp computers or on other operating systems. However it serves as an example of an operational handler for the combination of a 4811 controller and the enclosed DMP controller both driving a single CRT display.

APPENDIX

The following thumb-in programs allow the two controllers to be operated for debugging.

I. DMP Connection (OUTPUT FROM SWITCH REG)

TA: Loc 75

TC: Loc 65

I/O DEVICE 20_{16}

DMP Device 5

Priority 9

LOC	CONTENTS	
65	FFFF	NEG WORD CNT
1000	E620	
1	0075	REG 2 TO TA
2	E600	
3	2000	DATA FRM SW TO MEM
4	4210	XFER INIT
5	4A30	INPUT STATUS
6	AB7F	TEST BIT 8
7	E700	
1008	1000	LOOP TO START

Preset registers:

1 to C000 XFER INIT CODE

2 to 2000 MEM LOC OF DATA

II. 4811 OUTPUT FROM SWITCH REG (High Speed Channel)

LOC	CONTENTS	
1000	ED20	
1	4002	
2	4128	select channel 2
3	ED30	
4	8033	
5	4139	INIT Channel 2
1006	4509	output Data
7	4949	input status
8	7648	
9	1007	TESTING bit 8
A	E700	
100B	1006	LOOP

III. 4811 INPUT WITH ECHO ON HIGH SPEED CHANNEL

LOC	CONTENTS	
1000	ED20	
1	4003	
2	4128	select channel 3
3	ED30	
4	8037	
5	4139	Initiate
1006	4959	Get status
7	7658	
8	1006	
9	4D49	
A	E700	Loop
100B	1006	

This report was done with support from the Department of Energy. Any conclusions or opinions expressed in this report represent solely those of the author(s) and not necessarily those of The Regents of the University of California, the Lawrence Berkeley Laboratory or the Department of Energy.

Reference to a company or product name does not imply approval or recommendation of the product by the University of California or the U.S. Department of Energy to the exclusion of others that may be suitable.

TECHNICAL INFORMATION DEPARTMENT
LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720