DAVE,
THIS IS YOUR COPY OF DRAFT!

HP/HYTYPE INTERFACE DESCRIPTION

Rodney Inefuku

The following write-up describes the operation of the HYTYPE interface and also provides programming information for the HP.

References:

(1) Model 12ØØ HYTYPE I Product Description (orange colored)

(2) Model 12ØØ HYTYPE I Printer Maintenance Manual Preliminary (blue colored)

(3) A Pocket Guide to Interfacing HP 21ØØ Computers

In general, the purpose of the HP/HYTYPE interface is to receive a 16-bit word from the HP 21ØØ mini-computer, hold it in its buffer until the HYTYPE printer has completely accepted the data contained within part of the 16-bit word, and make a request to the HP for another word when the HYTYPE printer is ready to accept more data.

The interface receives the 16-bit word via the HP IOBO bus. When the word is stored, the bits are then assigned to do the following: a) HP data bits $IOBO_{\emptyset}$ thru $IOBO_{1\emptyset}$ are assigned to HYTYPE data and are sent through a cable to the HYTYPE printer. These bits correspond to the HYTYPE D1 thru D1Ø24 data bits. b) HP data bits $IOBO_{11}$ thru $IOBO_{13}$ are used to create control signals for the HYTYPE. c) HP data bits $IOBO_{14}$ and $IOBO_{15}$ are used to create enable - interrupt signals so that interrupt requests to the HP are made automatically as long as desired. Detailed discussions will be given later.

Storing of the data from the IOBO bus into the interface buffer is accomplished when the interface is addressed by (LSCL·IOG·LSCM) together with the IOO signal.

Parity is computed from the 11-bit word and sent over the cable to the HYTYPE interface. The HYTYPE interface print 1 of 2 shows that parity is computed from the rcv'd 11 data bits together with the rcv'd parity bit. If there is any error, the status is sent back to the HP interface while the strobes to the HYTYPE are disabled at the HYTYPE interface to prevent the printer from acting on the erroneous information. Parity error is gated back only when the HYTYPE interface receives either the <u>CHARACTER</u>, <u>HORIZONTAL</u>, or the <u>VERTICAL</u> strobes since the data lines will have "settled" when any of the strobes arrive. If parity error does occur, the error status is latched (in an unconventional manner) into a D flip-flop at the HP interface by "CLEARING" the flip-flop. The D flip-flop is used this way (see schematic, HP interface sheet 1 of 2) because parity sets the flip-flop without having to depend on any clock signal where on the other hand the IOI clock is used to reset the flip-flop ~~since the IOI clock is used to~~ because it looks at the status bits. In other words, after the HP looks at the IOBI signals for status, the D flip-flop is "RESET" by the trailing edge of the (HYTYPE address·IOI) signal -- the trailing edge meaning, "I have finished looking at the IOBI status bits, and so parity error latch can be cleared." The HP sees this parity error two ways: $IOBI_{15}$ which gives error in general; and $IOBI_6$ which is parity error. A detailed discussion of bits $IOBI_{15}$ and $IOBI_6$ will be given later.

Periodically the HP will look at the following signals to see if everything is "A-okay": $IOBI_{15}$ (error), $IOBI_4$ (printer ready), $IOBI_5$ (check), ? $IOBI_6$ (parity error), $IOBI_7$ (paper out), along with $IOBI_{\emptyset}$ (ready for character), $IOBI_1$ (ready for carriage control), and $IOBI_2$ (ready for paper feed). This takes place when the HYTYPE address is given together with signal IOI.

Bits $IOBO_{11}$ and $IOBO_{12}$ are dedicated in creating either the character, horizontal (carriage motion), vertical (paper feed), or the ribbon lift clock (refer to HP interface print 1 of 2). The bit combinations are decoded in the interface and are assigned in the following manner:

IOBO bits	12	11	
	Ø	Ø	character strobe
	Ø	1	horizontal strobe
	1	Ø	vertical strobe
	1	1	ribbon lift clock

move ribbon lift to 00 and then up to ccw. To p.6

A 1.2 μs one-shot multivibrator is used to create a 1.2 μs strobe pulse and a 25Ø ns one-shot multivibrator is used to delay it for 25Ø ns so that data lines will have settled at the HYTYPE when the strobe appears. As shown on the HP interface print 1 of 2, the sequence of delaying and developing the strobe signal begins when the clock to the buffer "goes away". The reason being that it is assumed the data is stored in the buffer by the trailing edge of the clock signal. The desired strobe signal is determined by the decoding of $IOBO_{11}$ and $IOBO_{12}$ which enables the strobe through to the HYTYPE. For more information on the required timing, refer to Fig. 3.2 on page 3-4 of the Product Description Manual for the HYTYPE.

Ribbon lift position is determined by a program instruction. The ribbon lift need only be set once and therafter will remain in that position until another ribbon lift instruction is given. Bit $IOBO_{13}$ is dedicated to ribbon lift position as follows:

$IOBO_{13}$	
Ø	down position
1	up position

A D flip-flop is used to store the ribbon lift information. The ribbon lift instruction thus requires the following:

$IOBO_{11} = 1$
$IOBO_{12} = 1$ Ribbon Lift

$IOBO_{13}$ = Either 1 or Ø depending on position desired

Note on HP schematic sheet 1 of 2 that the clock to the D flip-flop is enabled when $IOBO_{11} = IOBO_{12} = 1$.

Reset signal is created by either POPI/O or CRS signals, and as seen on the schematic it resets just about everything. Also, a 1.3 μs reset pulse is created for the HYTYPE by a one-shot multivibrator triggered by RESET signal.

As was mentioned before, the following status signals from the HYTYPE, are assigned corresponding IOBI bits, and are looked at by the HP:

printer ready	$IOBI_4$
check	$IOBI_5$
parity error	$IOBI_6$
paper out	$IOBI_7$
character ready	$IOBI_0$
horizontal ready	$IOBI_1$
vertical ready	$IOBI_2$

good bit to pick from my standpoint

you must be kidding

The $IOBI_{15}$ bit is designated as "error in general" and is an "oring" of printer ready + check + parity error + paper out signals -- which obviously means "if you got any one of these, you got an error!" This signal was created as a convenience for the programmer so that he need only look at the sign bit of IOBI to see if there is any error. If so, he can then go back to see which one.

In discussing interrupts, let me first point out that although there are three types of interrupts, the HP, via its interrupt request, will recognize just "<u>INTERRUPT</u>" in general. The distinction between the three is made in the HP interface. The three types of interrupts are character, horizontal, and vertical. Any interrupt request is determined by an 'ENABLE" signal. Bits $IOBO_{14}$ and $IOBO_{15}$ are assigned to the "ENABLE" signals. They are decoded as follows:

IOBO bits	15	14	
	Ø	Ø	no interrupt request
	Ø	1	Char Int. enable
	1	Ø	Hor. Int. enable
	1	1	Vert. Int. enable

The Product Description Manual for the HYTYPE on pg. (3 - 4) states that a carriage motion strobe (vertical or horizontal) may follow a character strobe 4ØØ ns later and vice-versa. The HP program allows the interface to make an interrupt request to the HP at approximately the same time that the strobe is being sent ot the HYTYPE because it will take a substantial amount of time (relative to the time requirement between successive strobes) for the HP to service that request. The principle on which an "INTERRUPT-ENABLE" is determined is that the interrupt-enable for instruction 'P' is given during instruction 'P-1'. In other words, the interrupt-enable for the present instruction be made by the previous instruction or the interrupt-enable for the next instruction is made by the present instruction.

As shown on the HP interface print 2 of 2, an interrupt-request is made by a 2ØØ ns pulse created by a one-shot multivibrator triggered by the appropriate interrupt-enable. On close examination of the interrupt-request circuit, one can see that the multivibrator is triggered almost immediately by the selected interrupt-enable when it is allowed to pass through by the trailing edge of the buffer clock (see HP INTERFACE sheet 1 of 2). ~~And~~ Then the next word would be loaded and executed by the HP interface without making an interrupt-request for the word after that. The reason is as follows: Consider that the D flip-flop for the interrupt-requests are turned "off" and "on" by their respective strobe and ready signals, and also that when a D flip-flop is turned "off" by its strobe, it in turn disables its corresponding interrupt-enable signal and keeps it from triggering the interrupt-request multivibrator. Now in the normal operation of the HYTYPE, a character instruction is usually followed by a displacement instruction and vise-versa. Hence, although word P (command?) in the interface will cause its strobe to turn off its D flip-flop, it normally will not affect word P + 1's enable. Thus word P + 1's interrupt-request is made almost immediately. Considering that word P + 2 is normally of the same type as word P, if the HP goes ahead and services the request for word P + 1 in the midst of not receiving a ready signal from the HYTYPE for the execution completion of the HYTYPE instruction given by word P, it must then await this 'ready' signal from the HYTYPE before continuing since the interrupt-enable to request for word P + 2 is in actuality controlled by this ready signal. The interrupt-enable for word P + 2 is enabled only when the ready signal returns form the HYTYPE and "clocks" its flip-flop into a "HI" state.

The circuits for the FLAG buffer, HTP flag, HTP IRQ, and the HTP CONTROL flip-flops are from the standard interface design offered by HP in its interface manual. An excellent description of these circuits is given in the manual.

Normally the FLAG buffer flip-flop is set by either HP I/O program instruction STF (set flag) or the 2ØØ ns pulse from the interrupt-request multivibrator.

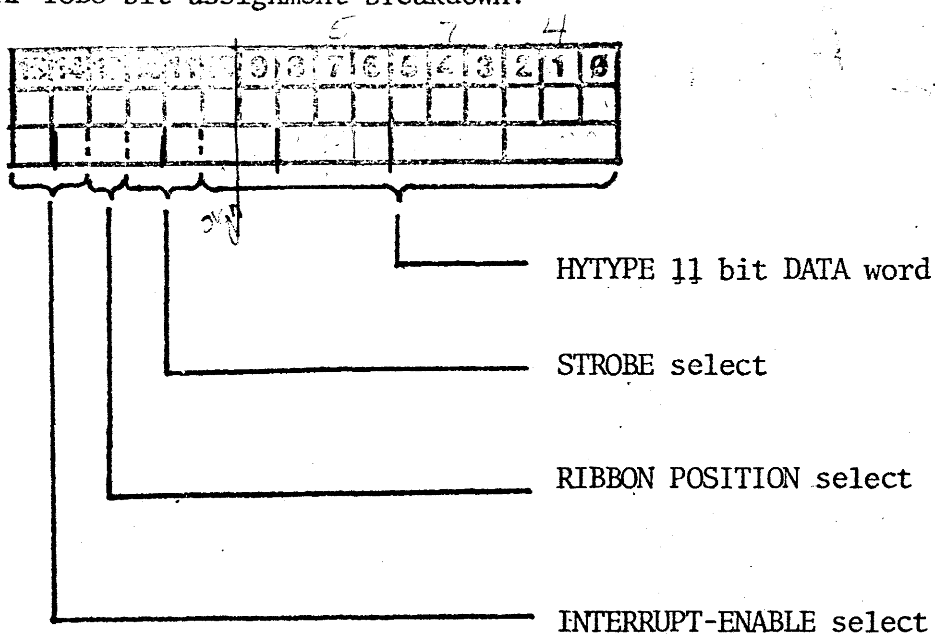
Instruction STF does so when the HYTYPE is initialized. In other words, instruction STF starts HYTYPE operation. Thereafter the interrupt-enable signals via the interrupt-request multivibrator automatically sets the FLAG buffer flip-flop so that program instruction STF need not be used repeatedly.

When the data from the last word is sent to the HYTYPE, no interrupt-enables are made ($IOBO_{14}$ = $IOBO_{15}$ = Ø) in this word since no interrupt-requests need to be made to the HP. The HYTYPE then returns to the "IDLE" state after the execution of its last instruction.

APPENDIX

The following is a synopsis of the bit assignments for the HP IOBO word.

HP IOBO bit assignment breakdown:



HYTYPE DATA WORD information:

The bit assignments for a character are in accordance with the SEVEN BIT ASCII CODE (Diablo modified). When representing an ASCII character, only the low order seven bits are used. The remaining bits are ignored.

When representing a carriage movement command, the ten low order bits designate the distance the carriage is to be moved in multiples of 1/60th of an inch. A value of 6 for each character or space moves the carriage at 10 characters/inch. the high order bit determines direction of carriage travel. A "zero" designates carriage movement to the right and a "one", movement to the left.

When representing a paper feed command, the ten low order bits designate the number of vertical positions, in multiples of 1/48th of an inch, that the paper is to be moved. The high order bit determines the direction of travel. A "zero" is to advance the paper upward and a "one" to move the paper downward.

STROBE select bit assignment:

IOBO bits 12	11	
∅	∅	character strobe
∅	1	horizontal strobe
1	∅	vertical strobe
1	1	ribbon lift clock

RIBBON POSITION bit assignment:

$IOBO_{13}$	
∅	down position
1	up position

INTERRUPT-ENABLE select bit assignment:

IOBO bits 15	14	
∅	∅	no interrupt request
∅	1	Char Int. enable
1	∅	Hor Int. enable
1	1	Vert Int. enable