PHILIPS

Electronic components and materials

Technical note 086

Introducing the Signetics 2655 PPI

The Signetics 2655 Programmable Peripheral Interface (PPI) is designed for use with the Signetics 2650 microprocessor. However, it is also ideally suited to microcomputer systems using other microprocessors, in polled or interrupt-driven environments. The 2655 PPI has 24 I/O pins, divided into three groups of eight: port A, port B and port C. There are seven modes of operation:

- static I/O
- strobed input
- strobed output
- bi-directional strobed I/O
- serial input
- serial output
- timer/event counter

This programmable interface circuit can thus be used in many differing applications to minimize the IC package count and provide a simple hardware interface between microcomputers and peripherals.

Circuit description

Figure 1 shows a block diagram of the Signetics 2655 Programmable Peripheral Interface.

All data and programming control transfers between the PPI and the microprocessor are accomplished via the data bus buffer, which connects the internal data bus to the microprocessor data bus.

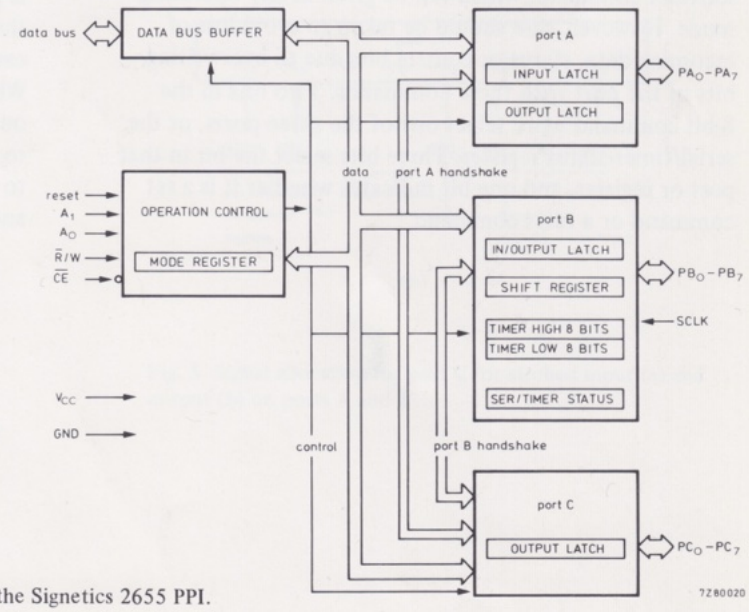


Fig. 1 Block diagram of the Signetics 2655 PPI.

Signetics

Access to the PPI is controlled by the Chip Enable ($\overline{CE}$) signal, while access to the various ports and registers is selected by address lines A_0 and A_1, and the Read/Write ($\overline{R}/W$) signal selects the direction of the data flow between the PPI and the microprocessor.

The operation modes of the PPI ports are selected by the mode word, written into the mode register by the microprocessor.

Three bits in the mode-word select the operating mode of port A, which can be:

- static I/O
- strobed input
- strobed output
- bi-directional strobed I/O

Three other bits in the mode-word select the operating mode of port B, which can be:

- static I/O
- strobed input
- strobed output
- serial input/timer
- serial output/timer

Note that the serial and timer modes operate simultaneously.

Port C needs no mode programming, since this port can only operate in the static I/O mode. However, the number of bits of port C that can be used for static I/O operation are limited when port A and/or port B claim pins of port C. This occurs during strobed mode operation when extra control signals are required. This is shown in the Fig. 1: the handshaking signals of port A and B can communicate with the microprocessor and the peripheral via port C.

Thus the programming of ports A and B can affect the use of the pins of port C, and the corresponding bits in the port C latch can be read by the microprocessor as information about the status of the strobed mode operation on ports A and/or B. A special register, which can be read by the microprocessor, is provided for the status of the serial/timer mode operation of port B.

Any bit of a specified port can be set or reset by a bit set/reset command, which can be given in any operating mode. However, care should be taken to avoid loss of incoming data, status or control bits due to overwriting bits of the port with these commands. Two bits in the 8-bit command word select one of the three ports, or the serial/timer status register. Three bits select the bit in that port or register, and one bit indicates whether it is a set command or a reset command.

Operating modes

Static I/O

Figure 2 shows the basic structure for a static I/O pin. The latch can be seen as having an open drain output with an internal pull-up resistor. Data written into the latch by the microprocessor drives the I/O pin; the latched data can also be read by the microprocessor. When a one is output on the I/O pin, it can be overruled by a zero input from a peripheral device. A one input by a peripheral device is similarly overruled by a zero output bit: the latch must therefore be loaded with a one prior to input operations. The unique internal circuitry in the 2655 automatically adjusts the I/O pin impedance levels for either input or output operation, and also provides protection against short circuits of the pin to either ground or V_{CC}.

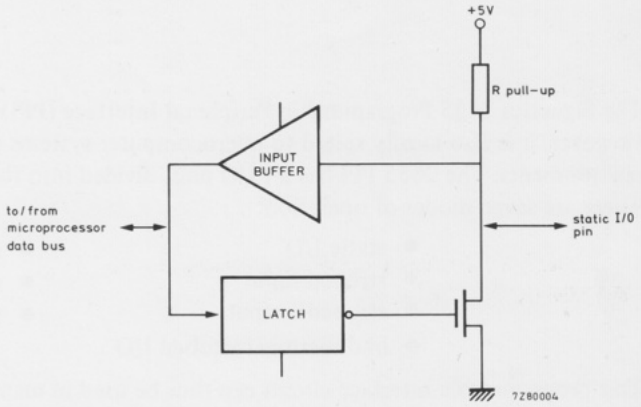


Fig. 2 Basic structure of a static I/O pin.

Each of ports A, B and C can operate in the static I/O mode, and any combination of input and output pins is possible. Each port can be written to, or read from by the microprocessor on an eight bit basis, while each bit can be individually set or reset by a bit set/reset command. When a port is used for a combination of inputs and outputs, bits corresponding to input pins must be OR-ed to 1 by a software mask when writting an eight bit word to the port. Similarly, a software mask can separate input and output data when reading from the port.

Strobed input and output

Strobed operation requires the exchange of control signals between the microprocessor and PPI and between the PPI and peripheral. This exchange of signals is called the handshake.

Figure 3 shows the signals required for the data input handshake and the handshake timing. Figure 4 shows the same details for data output.

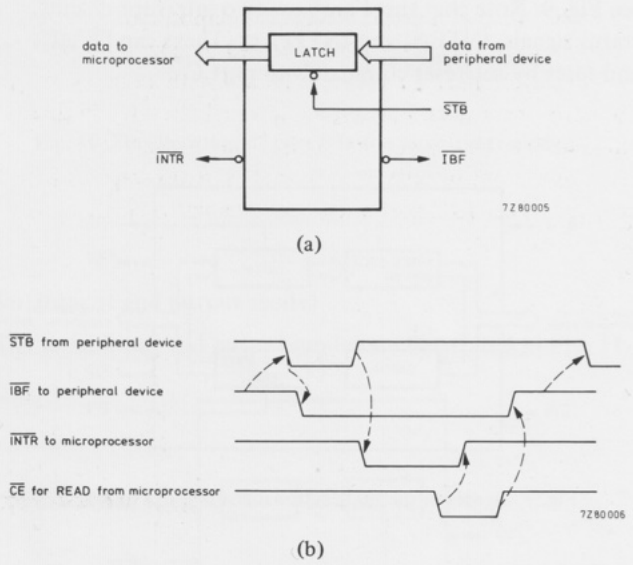


Fig. 3 Strobed input signals (a), and handshake timing (b).

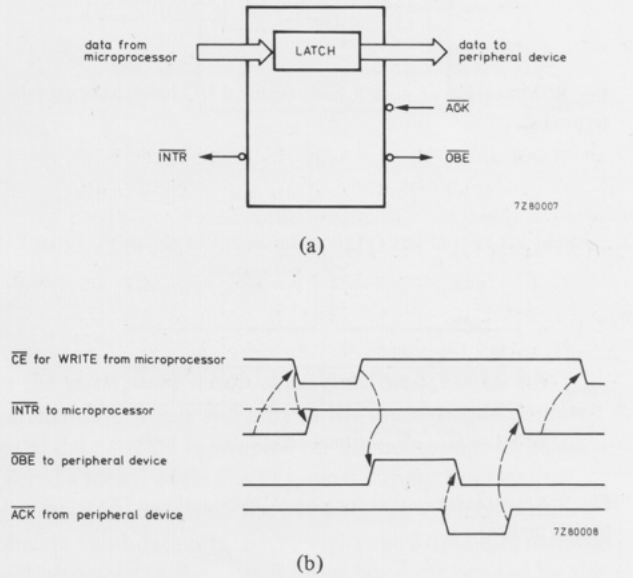


Fig. 4 Strobed output signals (a), and handshake timing (b).

$\overline{STB}$ Strobe, enables the input latch to accept the data from the peripheral device.

$\overline{IBF}$ Input Buffer Full, informs the peripheral device that its data has been accepted by the input latch, but has not yet been read by the microprocessor.

$\overline{INTR}$ Interrupt Request, *for an input handshake* informs the microprocessor that new data has been written into the input latch.
for an output handshake informs the microprocessor that the data in the output latch has been accepted by the peripheral device.

$\overline{OBE}$ Output Buffer Empty, informs the peripheral device that the microcomputer has not yet written new output data into the output latch.

$\overline{ACK}$ Acknowledge, indicates that the peripheral devices has accepted the data from the PPI latch.

Because there are insufficient pins on the package to allocate a pin to each handshake signal, some of the port C static I/O pins must be sacrificed. This means that when port A operates in the strobed modes, three of the static I/O pins of port C are used for the handshake signals. Similarly, port B strobed modes require three pins of port C, which means that if port A and/or port B operate in the strobed mode, port C has only 2 or 5 pins available for static I/O. Figure 5 shows the signal allocation of the pins of port C for strobed input/output on ports A and B.

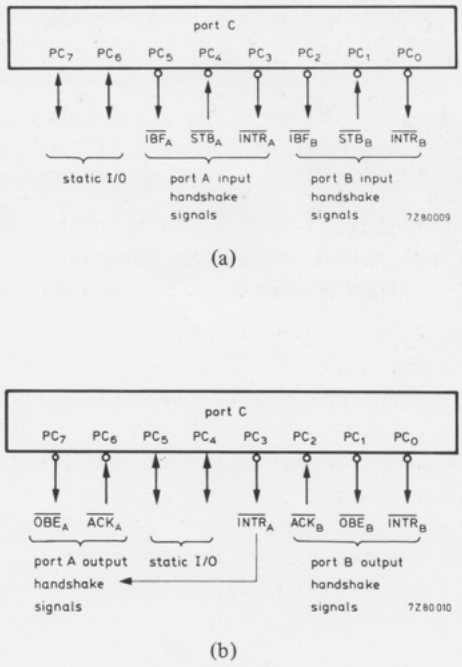
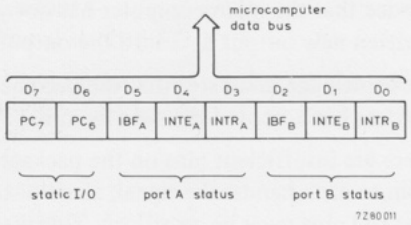


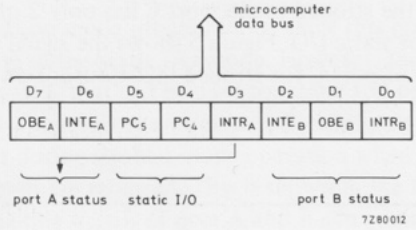
Fig. 5 Signal allocation for port C for strobed input (a) and output (b) on ports A and B.

The microcomputer can read the status of the handshake by reading the port C register. Figure 6 shows the meaning of the individual bits of the register. The INTE (interrupt enable) bits can be set or reset to enable/disable the INTR (interrupt request) signals at the pins of port C.

All pins of the 2655 PPI that are used for INTR signals are of an open-drain output type, so that they can be tied together to provide a single interrupt request signal. The microcomputer can then identify which signal has caused the composite interrupt request by examining the handshake statuses.



(a)



(b)

Fig. 6 Status information from port C register to microcomputer.

Bi-directional strobed I/O

Only port A can operate in this mode, which is illustrated in Fig. 7. It is a combination of the strobed input and output modes, with the interrupt request for input and output as one signal. Thus, this operation mode requires five pins of port C for its handshake signals. If port B is operating in the strobed input or output mode, all pins of port C are thus used for the handshake signals. Figure 8 shows the signal allocation. The microcomputer can read the status of the handshake by reading the port C register, see Fig. 9. Note that there are now two interrupt enable status signals, $INTE_{IN}$ and $INTE_{OUT}$. These can be set and reset by set/reset commands to port C.

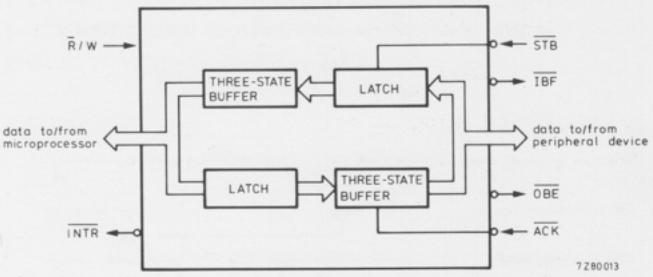


Fig. 7 Signals in the bi-directional strobed I/O mode.

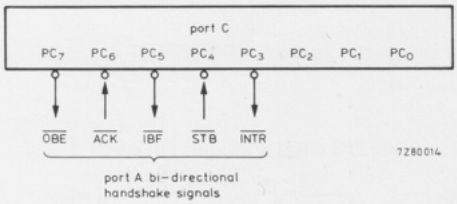


Fig. 8 Allocation of port A bi-directional I/O handshake signals to port C.

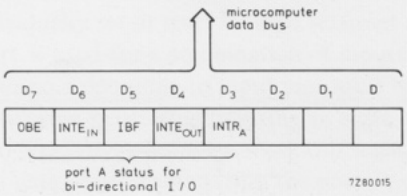


Fig. 9 Status information for port A bi-directional I/O mode in port C register.

Serial/timer mode

Only port B can be programmed for the serial/timer mode. The timer mode always operates simultaneously with either the serial input mode or the serial output mode. Figure 10 shows the use of port B pins in the serial/timer mode.

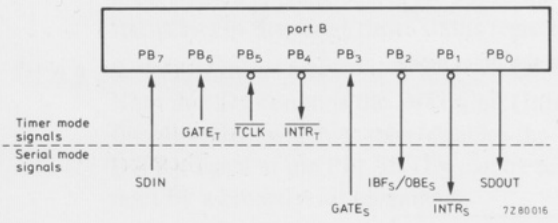


Fig. 10 Pin allocation of port B in the serial/timer mode.

Serial input and output modes

The serial input and output modes are illustrated in Fig. 11.

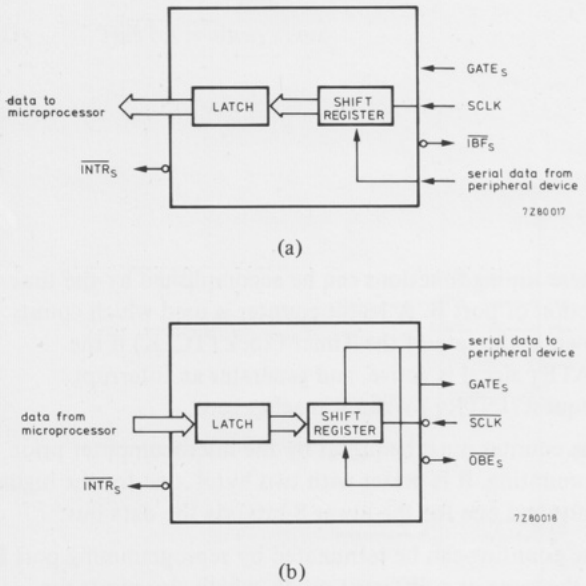


Fig. 11 Principle of the serial input (a) and output (b) modes.

In the serial input mode, serial data is converted into parallel data words and sent to the microprocessor. The serial data stream is sampled on the rising edge of SCLK (Serial Clock), while GATES must supply the framing information, on which the PPI will know which serial bits are valid data bits. Every eight valid data bits received will be placed in the port B input latch, to be read by the microprocessor. The handshaking signals are similar to those of the strobed modes: INTRS (Interrupt Request-serial) tells the microprocessor that a data word has been received and placed into the port B input latch. IBFS (Input Buffer Full-serial) tells the peripheral device that converted data is available for the microprocessor, but has not yet been read.

If GATES is de-activated before a full-8-bit serial word is received, the contents of the shift register at that moment are transferred to the port B latch. This can be used to convert serial data words of less than 8 bits into parallel data words.

The serial output mode is used for converting parallel data bytes from the microprocessor to serial form. Data placed in the port B latch by the microprocessor is transferred to the shift register and serialized at the rate of the SCLK input signal.

The GATES signal serves two functions. As an output, it frames valid data transfers. It is high when a "new" data byte is being serialized and low when a previous data byte is re-transmitted due to failure of the CPU to load the port B latch in time. If this output is pulled low while it is in the high output state, the subsequent actions depend on the state of the holding register at the time the output is pulled low:

(a) If a new data byte had been loaded into the holding register by the CPU, the byte is transferred to the shift register and serialized when GATES is released, GATES will go high to indicate transmission of this new data.

(b) If the new data byte is loaded after GATES is pulled low, the data transmitted when GATES is released will be the remainder of the previous byte. GATES will remain low during this time. Following this, the new data will be output with GATES high to indicate valid data transmission.

This technique can be used to control the number of bits that are serialized in each byte. GATES must remain low for at least one SCLK cycle.

The serial mode uses five pins of port B: one for Serial Data Out (SDOUT), one for Serial Data In (SDIN), and three for the handshaking signals INTRS, GATES and IBFS/OBES. The Serial Clock uses a dedicated pin.

A special status register which can be read by the microprocessor is provided for the serial/timer mode.

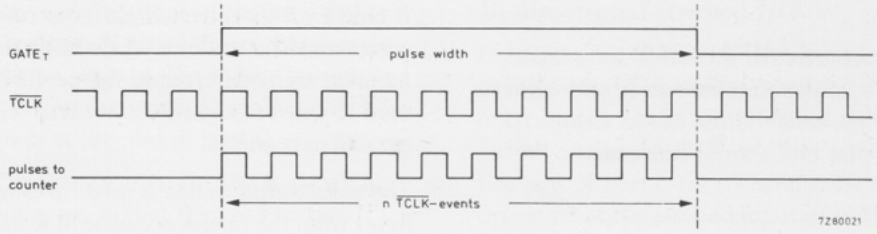


Fig. 12 Pulse width measurement in the timer mode.

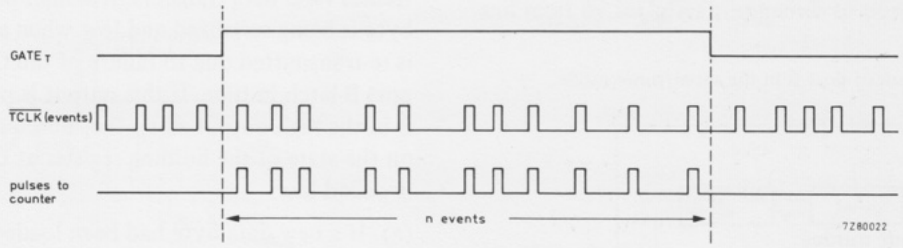


Fig. 13 Event counting in the timer mode.

Timer mode

The timer mode provides the 2655 PPI with the capability to perform *pulse width measurement, event counting, delay counting,* and *terminal event counting.*

- *Pulse width measurement*
  The time between the rising and the falling edges of the $GATE_T$ signal can be measured by counting the number of clock pulses of a constant frequency source, fed to the $\overline{TCLK}$ input, as shown in Fig. 12.
- *Event counting*
  The number of events, occuring during the time that the $GATE_T$ signal is active can be counted, when every event causes a pulse at the $\overline{TCLK}$ input, as shown in Fig. 13.
- *Delay counting*
  The counter is loaded by the microprocessor with a starting value, and counts down on the $\overline{TCLK}$ pulses (when $GATE_T$ is active), until the counter reaches zero, upon which an $\overline{INTR}_T$ (Interrupt Request-timer) is generated to interrupt the microprocessor. During the delay-time, between the loading of the counter and the interrupt Request, the microprocessor can execute other programs.
- *Terminal event counting*
  This is event counting, until a certain number of events have occurred. This is accomplished by loading the counter with a number, after which it will count down on $\overline{TCLK}$ pulses (events), until the counter reaches zero. This generates an $\overline{INTR}_T$ signal, upon which the microprocessor may take action.

These timing functions can be accomplished by the timer section of port B. A 16-bit counter is used which counts down at the rate of the Timer Clock ($\overline{TCLK}$) if the $GATE_T$ signal is active, and generates an Interrupt Request ($\overline{INTR}_T$) when it reaches zero.

The counter must be preset by the microcomputer prior to counting. It is preset with two bytes, one for the higher 8-bits and one for the lower 8-bits, via the data bus.

The counting can be terminated by reprogramming port B to the same or a different mode, which also resets the $\overline{INTR}_T$ signal.

The timer mode uses three pins of port B: one for $\overline{TCLK}$, $GATE_T$, and one for $\overline{INTR}_T$.

The serial/timer status register

The serial/timer status register bits can be divided into four groups as shown in Fig. 14.

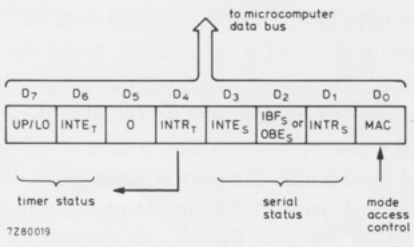


Fig. 14 Information in the serial/timer status register.

– D_0 *Mode Access Control* (MAC)
The serial mode and the timer mode share port B and the serial/timer status register. Access to either the port B latch (serial) or the timer registers is controlled by the MAC bit. This bit also controls which mode is to be programmed and also resets the appropriate status bits in the serial/timer status register.

– $D_{1,2,3}$ *Serial I/O status bits*
Note that D_3 contains the $INTE_S$ bit (Interrupt Enable-serial), which enables/disables the $INTR_S$ signal at pin PB_1. $INTE_S$ can be set/reset by a bit set/reset command.

– $D_{4,6,7}$ *Timer status bits*
Note that D_6 contains the $INTE_T$ (Interrupt Enable-timer), which enables/disables the $\overline{INTR}_T$ signal at pin PB_4.

D_7 selects which half of the timer (*upper or lower*) is accessed, during read/write from/to the timer. D_7 is automatically toggled each time the timer is accessed. D_6 and D_7 can be set/reset by a bit set/reset command.

– D_5 This bit is always zero.

Figure 15 shows an example where the 2655 PPI is programmed to have one 8-bit bi-directional strobed I/O port (port A), a serial output port and a timer (port B), and 3 static I/O pins, which are left on port C.

Note that all pins on the 2655 which are programmed for $\overline{INTR}$ have an open drain output configuration, so that they can be OR-tied together, with an external pull-up resistor.

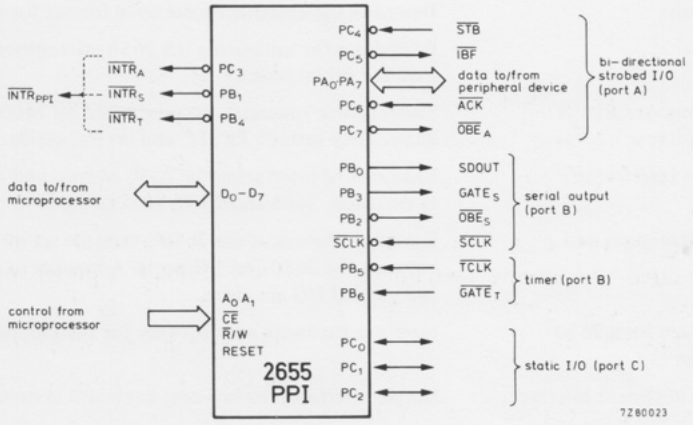


Fig. 15 Example of multi-mode programming of the 2655 PPI.

Related 2650 publications

no.	title	summary
AS50	Serial Input/Output	Using the Sense/Flag capability of the 2650 for serial I/O interfaces.
AS51	Bit & Byte Testing Procedures	Several methods of testing the contents of the internal registers in the 2650.
AS52	General Delay Routines	Several time delay routines for the 2650, including formulas for calculating the delay time.
AS52	Binary Arithmetic Routines	Examples for processing binary arithmetic addition, subtraction, multiplication, and division with the 2650.
AS54	Conversion Routines	• Eight-bit unsigned binary to BCD • Sixteen-bit signed binary to BCD • Signed BCD to binary • Signed BCD to ASCH • ASCII to BCD • Hexadecimal to ASCII • ASCII to Hexadecimal
AS55	Fixed Point Decimal Arithmetic	Methods of performing addition, subtraction, multiplication and division of BCD numbers with the 2650.
SP50	2650 Evaluation Printed Circuit Board (PC1001)	Detailed description of the PC1001, an evaluation and design tool for the 2650.
SP51	2650 Demo System	Detailed description of the Demo System, a hardware base for use with the 2650 CPU prototyping board (PC1001 or PC1500).
SP52	Support Software for use with the NCSS Timesharing System	Step-by-step procedures for generating, editing, assembling, punching, and simulating Signetics 2650 programs using the NCSS timesharing service.
SP53	Simulator, Version 1.2	Features and characteristics of version 1.2 of the 2650 simulator.
SP54	Support Software for use with the General Electric Mark III Timesharing System	Step-by-step procedures for generating, editing, assembling, simulating, and punching Signetics 2650 programs using General Electric's Mark III timesharing system.
SP55	The ABC 1500 Adaptable Board Computer	Describes the components and applications of the ABC 1500 system development card.
SS50	PIPBUG	Detailed description of PIPBUG, a monitor program designed for use with the 2650.
SS51	Absolute Object Format	Describes the absolute object code format for the 2650.
MP51	Initialization	Procedures for initializing the 2650 microprocessor, memory, and I/O devices to their required initial states.
MP52	Low-Cost Clock Generator Circuits	Several clock generator circuits, based on 7400 series TTL, that may be used with the 2650. They include RC, LC and crystal oscillator types.
MP53	Address and Data Bus Interfacing Techniques	Examples of interfacing the 2650 address and data busses with ROMs and RAMs, such as the 2608, 2606 and 2602.
MP54	2650 Input/Output Structures and Interfaces	Examines the use of the 2650's versatile set of I/O instructions and the interface between the 2650 and I/O ports. A number of application examples for both serial and parallel I/O are given.
TN 064	Digital cassette interface for a 2650 microprocessor system	Interface hardware and software for the Philips DCR digital cassette drive.
TN 069	2650 Microprocessor keyboard interfaces	Simple interfaces for low-cost keyboard systems.
TN 072	Introducing the Signetics 2651 PCI Terminology and operation modes	Description of the 2651 Programmable Communications Interface IC.
TN 083	Using the Signetics 2651 PCI with popular microprocessors	Simple hardware interfaces to use the 2651 Programmable Communications Interface with various microprocessors.
TN 084	Using seven-segment LED displays with the 2650 microprocessor	Interfaces for single and multi-digit LED displays.
TN 085	Cyclic redundancy check by software	A short routine to encode and decode CRC check characters for the 2650.

Printed in The Netherlands Date of release: 29 May 1978 T.D. 9399 500 08601