FACIT SP1

Facit standard interface for parallel data transfer

Technical description

Part 2 - Applications

TABLE OF CONTENTS

Sections 1 through 6 of this technical description entitled "FACIT SP1, Facit standard interface for parallel data transfer, Technical description" (document SP1.15.01.Eng.10M.09.71) presents the specifications for the Facit standard interface SP1.

FACIT SP1

Facit standard interface for parallel data transfer

Technical description

Part 2- Applications

7 SIGNALS AO AND SO

Signal AO from an acceptor and signal SO from a source indicate, by means of a logic 1, that the device in question is operable. The term "operable" as used here means that all of the requisite conditions for data transfer by the device in question (such as power on, tape in position, cover closed etc.) are fulfilled.

AO and SO can be used to prevent the inadvertent transfer of characters in connection with the turning on and off of power. If AO or SO are kept at 0 from power-on time until control signal AC or SC has stabilized, AC and SC will be interpreted as 0.*) This also holds true if AO or SO are set to 0 at power-off time prior to the start of any variations that may occur in AC or SC. Examples are shown in Figs. 7.1 and 7.2

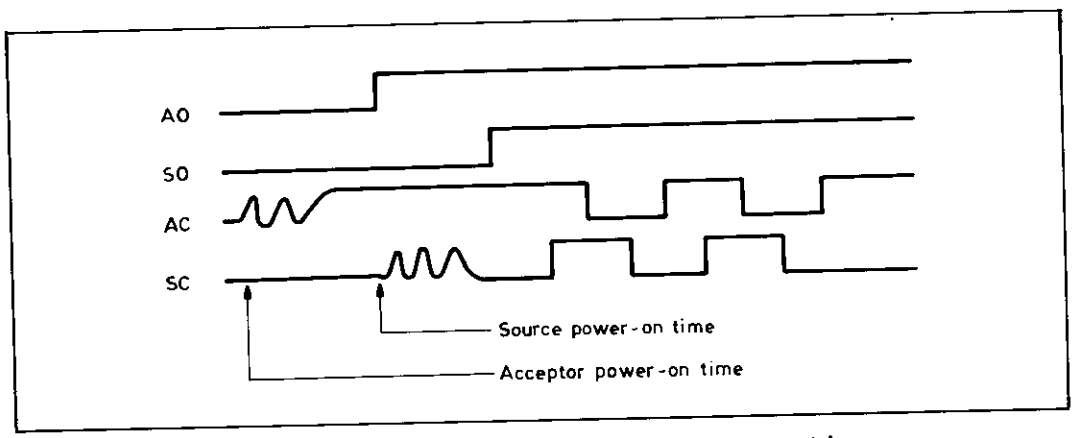


Fig. 7.1 Example of what happens at power-on time

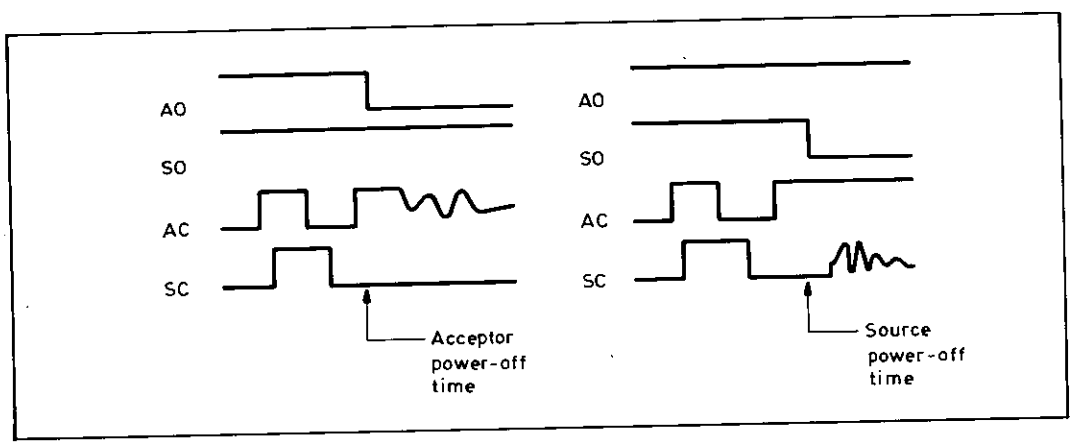


Fig. 7.2 Example of what happens at power-off time

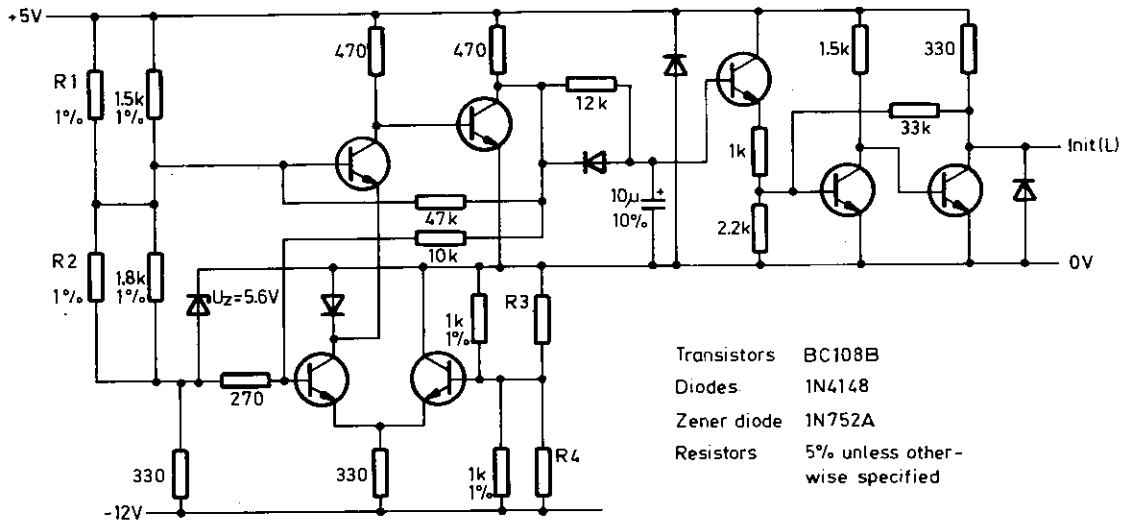
*) See sections 3.3.2 and 3.4.2 in the first part of this technical description.

If A0 or S0 is controlled by supply voltages, non-intentional voltage drops can be detected in time and A0 and S0 can be set to 0 before AC or SC is affected. Fig. 7.3 shows one example of a circuit in which A0 and S0 are voltage-controlled. Signal Init(L) (Initialize(L)) is determined by the supply voltage levels in the power supply, and it is sent from the power supply to the connected devices where it controls A0 and S0. If the +5 V supply sent to a device fails, but remains present in the power supply - the A0 or S0 signal from the device will drop to 0 in spite of the fact that Init(L) remains at the high level.

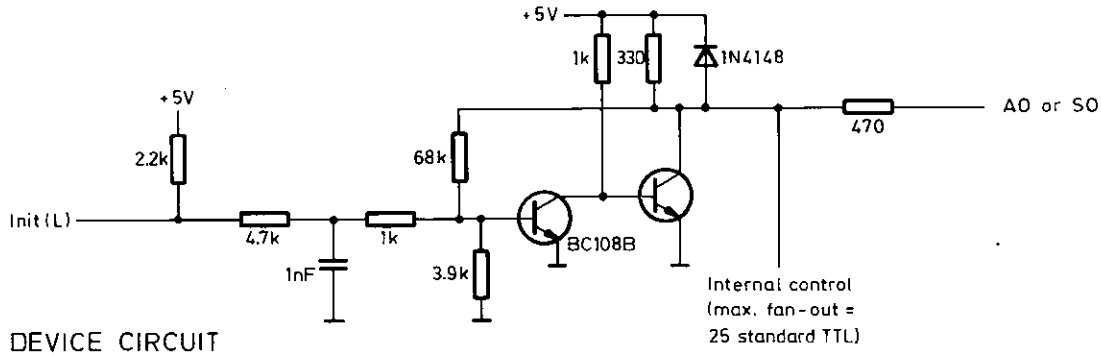
Fig. 7.4 presents the situations that can arise if A0 or S0 becomes 0 while the transfer of characters is in progress. See section 12.1 for a more detailed discussion of the initialization circuit.

A device shall use A0 or S0 to signal that it has entered a state (end of tape for example) in which the transfer of characters is no longer possible. Fig. 7.4 shows that if A0 or S0 is set to 0 when both AC and SC are 0, neither the transmission of characters from the source nor the AM message from the acceptor will be disturbed. As long as A0 or S0 is 0, the device in question cannot participate in any transfer of characters. If a condition that makes continued data transfer impossible is reported using AM or a message sent via the data wires instead of by setting A0 or S0 to 0, corrective commands or referrals can still be transferred to the reporting device.

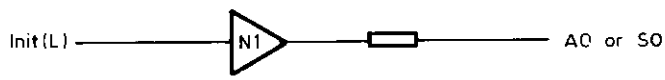
In a system comprising several devices joined to each other via a communication system, signals A0 and S0 can be utilized for ascertaining whether or not the devices are operable. This ascertainment does not involve the transfer of a character to or from the device in question, and it can be carried out regardless of the working state of the device.



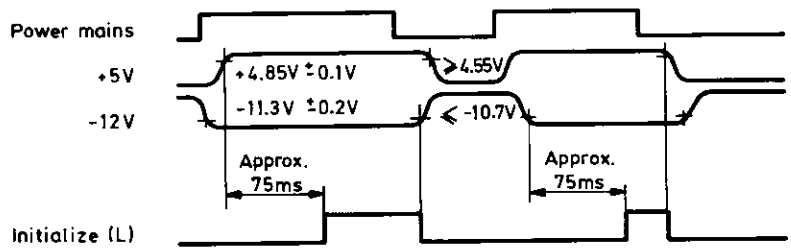
POWER SUPPLY CIRCUIT



DEVICE CIRCUIT



Symbol for device circuit



The ohmage of R1 or R2 shall be determined by trials conducted to provide a power-on ascertainment level of +4.85 V for the +5 V supply. The ohmage of R3 or R4 shall be determined by trials conducted to provide a power-on ascertainment level of -11.3 V for the -12 V supply.

If the power-on ascertainment levels are adjusted to nominal values and the internal ambient temperature ranges from 0 to 60° C, the power supply circuit provides:

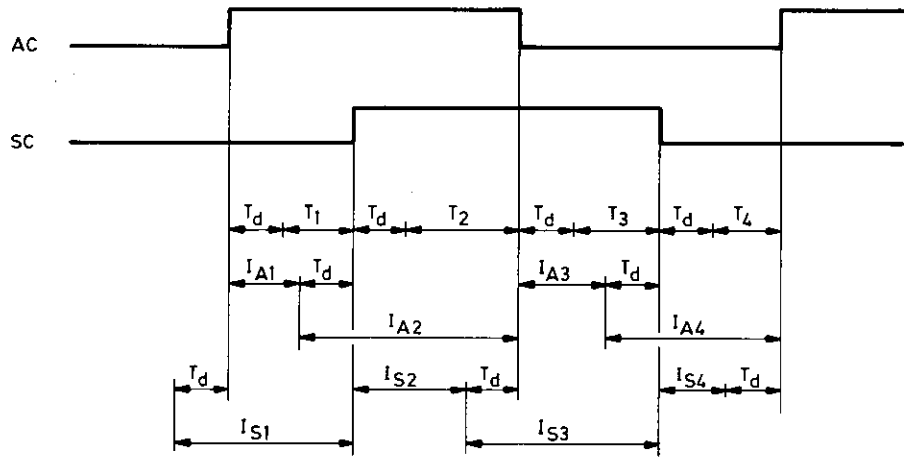
+5 V Power-on ascertainment level: +4.85 ± 0.1 V, power-off ascertainment level: ≥+4.55 V

-12 V Power-on ascertainment level: -11.3 ± 0.2 V, power-off ascertainment level: ≤-10.7 V

Init(L) is delayed about 75 ms in connection with power turn-on, on condition that the +5 V supply was below its power-off ascertainment level for at least 1 ms - and on condition that the -12 V supply was above its power-off ascertainment level for at least 1 ms. High level for Init(L) is ≥3.7 V when at least 1 device is connected. Low level for Init(L) is ≤0.5 V. This circuit can be loaded with up to 15 devices.

The receiving circuits for the Init(L) signal in the devices have a hysteresis that is ≥0.3 V. Levels ≥2.3 V are interpreted as Init(L)=high. Levels ≤0.8 V are interpreted as Init(L)=low.

Fig. 7.3 Example of circuit in which AO and SO are voltage-controlled



S designates source

A designates acceptor

T_1 and T_3 designate operation times in source

T_2 and T_4 designate operation times in acceptor

T_d designates cable delay

T_1, T_2, T_3 and T_4 include the intervals needed to detect a change in an incoming control signal.

I_{A1}, I_{A2}, I_{A3} and I_{A4} divide the transfer cycle into four intervals referred to in the discussion of AO.

I_{S1}, I_{S2}, I_{S3} and I_{S4} divide the transfer cycle into four intervals referred to in the discussion of SO.

AO becomes 0 during

I_{A1} - Detected by S before SC is set to 1. There may not be time for S to receive the AM message, if any, sent out by A.

I_{A2} - Interpreted by S as "AO becomes 0" when AC and SC are 1. S shall be designed so that this is not interpreted as an acknowledgment from A. (See second paragraph in section 4.1.3 in the first part of this technical description.)

I_{A3} - Interpreted by S as "AO becomes 0" after A has correctly acknowledged the received character. S is permitted to set SC to 0.

I_{A4} - Occurs during the interval while AC and SC are 0. Does not require any action in S.

SO becomes 0 during

I_{S1} - Interpreted by A as "SO becomes 0" while AC = 1 and SC = 0. Requires no action in A.

I_{S2} - Interpreted by A as "SO becomes 0" while AC and SC are 1, i.e. while a character is being fed out from S. There may not be time for A to receive the character that is sent out by S.

I_{S3} - Interpreted by A as "SO becomes 0" after AC, as an acknowledgment of the received character, is set to 0. It is not certain that S will have had time to interpret the acknowledgment from A.

I_{S4} - Occurs during the interval while AC and SC are 0. Does not require any action in A.

Fig. 7.4 Events occuring in a source when AO is set to 0 and in an acceptor when SO is set to 0 during different intervals throughout the transfer cycle

8 SIGNALS AC AND SC

Control signals AC and SC control character transfer between a source and an acceptor. These signals operate using a method of "handshaking" wherewith a level change in the control signal from one device is always followed by a level change in the control signal from the other device.

Each change in the level of the control signal received from device I means that the device I has interpreted the preceding change of the control signal from device II. It also means that device I has generated any answering message on its output lines. Control signal changes mark the changeover between two adjoining phases of a transfer cycle. Each time a control signal changes, the initiative is transferred from the signalling device to the other device. Normally, a device shall answer as quickly as possible, even though no timing requirements are stipulated for handshaking.

The character transfer rate is determined by three factors:

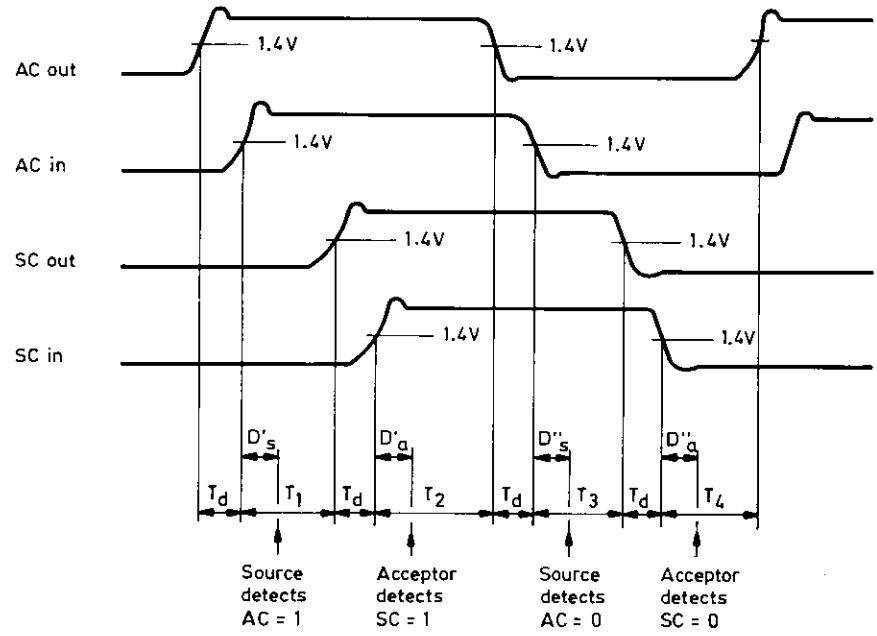
a) total time needed for the devices to carry out the different phases of a complete transfer cycle,

b) the amount of signal delay in the device circuits adjoining the interface and

c) the delay in the cable that joins the devices.

When a system comprises several devices served in sequence, it is vital that the maximum permissible delay times be specified so that the waiting times within the system can be limited to acceptable values.

Fig. 8.1 defines the time intervals that make up the transfer cycle and thus determine the transfer rate.



AC out - Signal AC at the acceptor connector

AC in - Signal AC at the source connector

SC out - Signal SC at the source connector

SC in - Signal SC at the acceptor connector

T_d - Cable delay, i.e. the interval elapsing from the time at which a signal enters one end of the cable until it reaches the other end of the cable.

T_1 - Interval elapsing from the time when AC changes from 0 to 1 until SC changes from 0 to 1. Measured at the source connector.

T_2 - Interval elapsing from the time when SC changes from 0 to 1 until AC changes from 1 to 0. Measured at the acceptor connector.

T_3 - Interval elapsing from the time when AC changes from 1 to 0 until SC changes from 1 to 0. Measured at the source connector.

T_4 - Interval elapsing from the time when SC changes from 1 to 0 until AC changes from 0 to 1. Measured at the acceptor connector.

D'_a - Interval needed by the acceptor to detect the changing of SC from 0 to 1.

D''_a - Interval needed by the acceptor to detect the changing of SC from 1 to 0.

D'_s - Interval needed by the source to detect the changing of AC from 0 to 1.

D''_s - Interval needed by the source to detect the changing of AC from 1 to 0.

$T = T_1 + T_2 + T_3 + T_4 + 4 \cdot T_d$(seconds) - total time needed to transfer one character.

$R = 1/T$ (characters/second) - transfer rate.

Fig. 8.1 Definitions of the various time intervals that make up a transfer cycle anc thus determine the transfer rate

The acceptor initiates a transfer cycle by setting AC to 1. Previously, the acceptor shall have ascertained that the source has indicated that the previous character transfer was completed by setting SC to 0. In addition, the acceptor shall have assigned AM the value that it wishes to transfer to the source, and the acceptor shall be prepared to accept a character from the source. When the source receives AC = 1, it shall verify that A0 is also = 1, thus making certain that the acceptor purposely set AC to 1. The source shall then receive AM and, possibly as directed by the value of AM, fetch the next character and send it out on the data wires (CA, P and D1 through D8). When all this has taken place, the source shall so indicate by setting SC to 1. When the acceptor receives SC = 1, the acceptor shall verify that S0 also = 1, thus making certain that the source purposely set SC to 1. Thereafter the acceptor shall accept the character on the data wires. When the acceptor has ascertained that SC is 1, it knows that the AM message has been received and it can permit AM to assume an arbitrary value. When the acceptor has accepted the character on the data wires it shall set AC to 0 as an acknowledgment, and start internal processing (if appropriate). When the source has ascertained that AC is 0, it shall, by way of concluding the transfer cycle, signal that it has received the acceptor's acknowledgment by setting SC to 0. Due to the fact that the source has ascertained that AC is 0, it can permit the data wires to assume arbitrary states. When the acceptor has ascertained that SC is 0, it knows that the source has also concluded the transfer of characters. Fig. 8.2 presents a detailed timing diagram that illustrates the sequence of the various events and their internal relationships.

If A0 is 0, the acceptor is not operable and AC shall be interpreted as 0 (see section 3.3.2 in the first part of this technical description). If S0 is 0, the source is not operable and SC shall be interpreted as 0 (see section 3.4.2 in the first part of this technical description). This means that a character transfer operation being carried out when A0 or S0 becomes zero is immediately interrupted. See Fig. 7.4 for clarification of the various situations that can therewith arise.

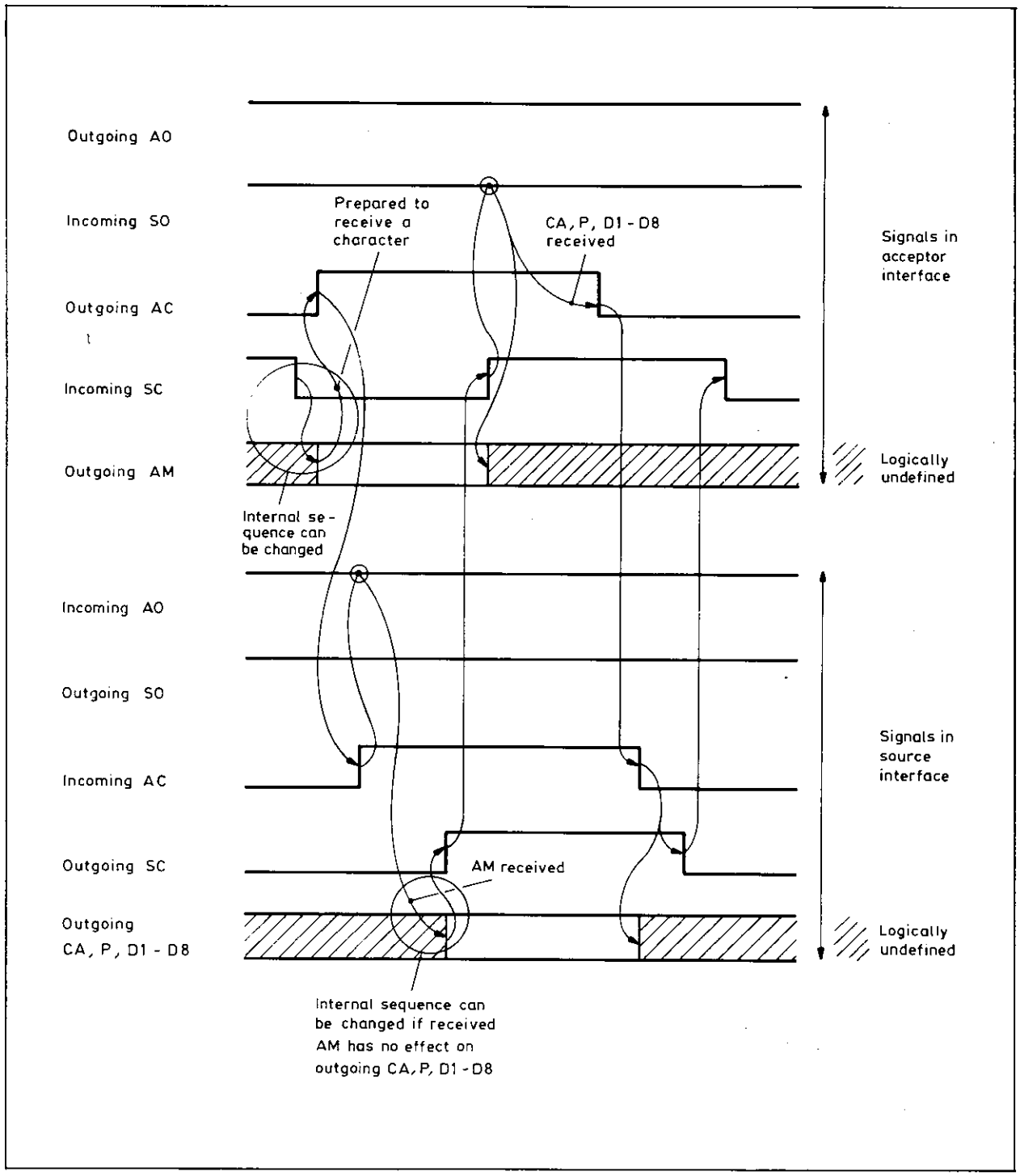


Fig. 8.2 Timing diagram of a transfer cycle

9 CODE AND PARITY FOR DATA WIRES

9.1 Code selection

The data wires consist of CA, P and D1 through D8. Characters sent from the source to the acceptor are encoded and carried on CA and D1 through D8. P carries the parity bit. Characters can represent:

- Control commands
- Status messages
- Data messages

Control commands govern the operation of the acceptor. Status messages report on certain conditions in the source. Data messages consist of the useful information or "payload" being handled by the devices.

Characters that are identified by certain devices and broken down into control commands, status messages or data messages are interpreted by other devices simply as data messages that are to be transferred, stored or processed.

Devices equipped with the Facit SP1 are used within systems. Two kinds of information are transmitted within a system:

a) the primary information or "payload" which the system is expected to transfer and

b) the information needed to operate the devices included in the system.

Sometimes a payload plus its associated device-operating information is carried through part of the system as a payload - a procedure that can be repeated on several levels.

The messages must be encoded in such a way that it will be possible to distinguish data messages and other messages on all levels. Standardized codes facilitate the use of devices equipped with the SP1 within different systems.

Coded characters consisting of up to nine bits can be transferred on CA and D1 through D8. D1 shall always transfer the least significant bit. Any of the data wires that are not used shall be kept at 0.

Standardized codes are available with 6, 7 and 8 bit characters. These codes contain control characters and graphic characters. Table 9.1 presents a summary of the international standardized codes and the standardized Swedish codes derived from them.

When deciding which code to use in a system it is advisable to first ascertain whether a standard code is available that is suitable for the primary information to be handled.

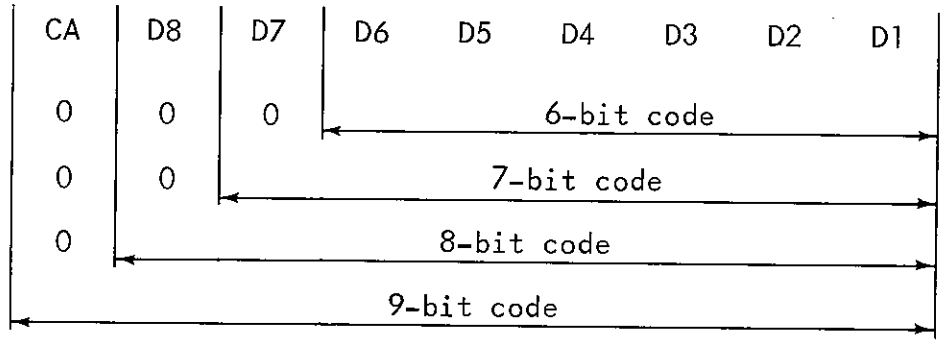
Primary information should be encoded using as few bits as possible, and the unused bits (having higher significance) shall be kept at 0.

Non-primary information can be encoded in two ways:

a) using the otherwise unused combinations of the primary-information bits and

b) using a larger number of bits.

This can be illustrated schematically as follows:



No standard has been adopted for a 9-bit code. 9-bit codes are divided into two parts: for one part CA = 0 and for the other CA = 1. In both parts, it is advisable to encode D1 through D8 in accordance with some standard 8-bit code if possible. In such case, CA = 1 can be used to indicate that D1 through D8 represent control or status information.

There is no standard for status messages. Conditions which are to be reported are determined by events that, as a rule, are not mutually exclusive and therefore must each be assigned an individual bit position in the status message.

When coded characters must be formed for control commands not included in any standard code, decoding can be facilitated by reserving one bit position for each function that is to be controlled. However, this arrangement occupies more space than multi-bit coding.

If individual-bit coding is not possible because of space considerations, functions that will never be called for simultaneously can be combined to form a group. Several groups can be formed in this manner for a single device. Functions can then be encoded one at a time from each group using a limited number of bits in the control command.

9.2 Standardized codes

A code having b bits can comprise 2^b combinations. These can be presented in a table having c columns and r rows, where

$$c \times r = 2^b$$

Coded characters are entered on the code table at the intersection representing the bit combination that has been assigned to them. Coded characters can then be designated by referring to their row and column if desired.

International standardization of codes used for information interchange is carried out within ISO and ECMA (see list below), and these two organizations maintain close contact with national standardization organizations as well as organizations such as CCITT operating within associated fields. In Sweden, this work is carried out jointly by two national organizations, namely SEK and SIS (see list below).

The 7-bit code specified in ISO R/646 and the ECMA-6 standard provides the basis for other information interchange codes.

Its code table contains 2^7 = 128 positions and has eight columns and 16 rows. Columns 0 and 1 contain control characters for:

- Transmission control
- Device control
- Text editing
- Code extension

Columns 2 through 7 contain graphic characters (graphics) some of which are alphanumeric.

Transmission control characters (controls) are used for transferring data. Rules for their use are presented in the ECMA-16 and ECMA-24 standards.

The ECMA-24 standard explains how code-independent transfer can be carried out using the DLE (data link escape) character.

Standard ECMA-16 for Basic Mode Control Procedures for Data Communication Systems Using the ECMA 7-Bit Code. May 1968.

Standard ECMA-24 for Code Independent Information Transfer. Dec. 1969.

ECMA: European Computer Manufacturers' Association

ISO: International Organization for Standardization

CCITT: Comité Consultatif International Télégraphique et Téléphonique (International Telegraph and Telephone Consultative Committee)

SEK: Svenska Elektriska Kommissionen (the Swedish National Committee of the International Electrotechnical Commission)

SIS: Sveriges Standardiseringskommission (Swedish Standards Institution)

Device control and text editing characters are described in ISO R/646 and ECMA-6. These characters control device operations that are evident to the user. After decoding is carried out in the acceptor, the combination of functions that perform the intended device operation is actuated.

Each of the three code extension characters: ESC (escape), SO (shift-out), SI (shift-in) changes the meaning of the word that follows it in accordance with pre-determined rules. See ISO/TC97/SC2 N562 and the ECMA-35 standard.

The 7-bit code was originally drawn up for transferring and editing text. Starting with the basic version, the user is permitted to define four of the device-controlling characters and to define his own graphic characters in up to ten predetermined positions in columns 2 through 7. Moreover, the user is permitted to choose between # and £ at position 2/3 and between ¤ and $ at position 2/4.

The basic version, without any changes, is called the international reference version. See Fig. 9.1.

Bits b_7	b_6	b_5	b_4	b_3	b_2	b_1	Row \ Column	0	0	0	0	1	1	1	1
								0	0	1	1	0	0	1	1
								0	1	0	1	0	1	0	1
								0	1	2	3	4	5	6	7
0	0	0	0	0			0	NUL	TC_7(DLE)	SP	0	@	P	`	p
0	0	0	1	1			1	TC_1 (SOH)	DC_1	!	1	A	Q	a	q
0	0	1	0	2			2	TC_2 (STX)	DC_2	"	2	B	R	b	r
0	0	1	1	3			3	TC_3 (ETX)	DC_3	#	3	C	S	c	s
0	1	0	0	4			4	TC_4 (EOT)	DC_4	¤	4	D	T	d	t
0	1	0	1	5			5	TC_5 (ENQ)	TC_8(NAK)	%	5	E	U	e	u
0	1	1	0	6			6	TC_6 (ACK)	TC_9(SYN)	&	6	F	V	f	v
0	1	1	1	7			7	BEL	TC_{10}(ETB)	'	7	G	W	g	w
1	0	0	0	8			8	FE_0 (BS)	CAN	(	8	H	X	h	x
1	0	0	1	9			9	FE_1 (HT)	EM	)	9	I	Y	i	y
1	0	1	0	10			A	FE_2 (LF)	SUB	*	:	J	Z	j	z
1	0	1	1	11			B	FE_3 (VT)	ESC	+	;	K	[	k	{
1	1	0	0	12			C	FE_4 (FF)	IS_4 (FS)	,	<	L	\	l	|
1	1	0	1	13			D	FE_5 (CR)	IS_3 (GS)	-	=	M	]	m	}
1	1	1	0	14			E	SO	IS_2 (RS)	.	>	N	↑	n	~
1	1	1	1	15			F	SI	IS_1 (US)	/	?	O	_	o	DEL

Fig. 9.1 International reference version of 7-bit code as specified in ISO R/646

The versions of the code selected by the user as described above are considerably more uniform than the codes that would have been drawn up if all users unable to use the basic version had been directed to draw up their own. Countries whose alphabets contain different national characters have found these tailored versions of the 7-bit code particularly useful.

8-bit codes can be formed on the basis of the 7-bit code by inserting the eighth bit before the most significant bit in the 7-bit code.

The 8-bit code table contains 16 columns and 16 rows. Columns 00, 01, 08 and 09 shall contain control characters. The other columns shall contain graphic characters.

To facilitate referral to different fields of the code table, ISO uses the following designations (see fig. 9.2):

Designation	Columns	
C0	00 - 01	Control characters
C1	08 - 09	Control characters
G0	02 - 07	Graphic characters
G1	10 - 15	Graphic characters

In order to extend the usability of an 8-bit code, the user is permitted to specify characters within fields C0, C1, G0 and G1.

8-bit codes in which fields C0 and G0 are identical to those in the basic version of the 7-bit code are of special interest.

In order to keep track of and limit the number of sets of control characters and graphic characters, users can have their character sets checked and registered by the ISO/TC97/SC2 secretariat. All codes so registered are made public. ISO and ECMA will draw up sets of control characters for different types of devices and have them registered. As more and more codes are registered, an assortment will become available. Among these, certain codes will stand out as the most suitable for use within different areas.

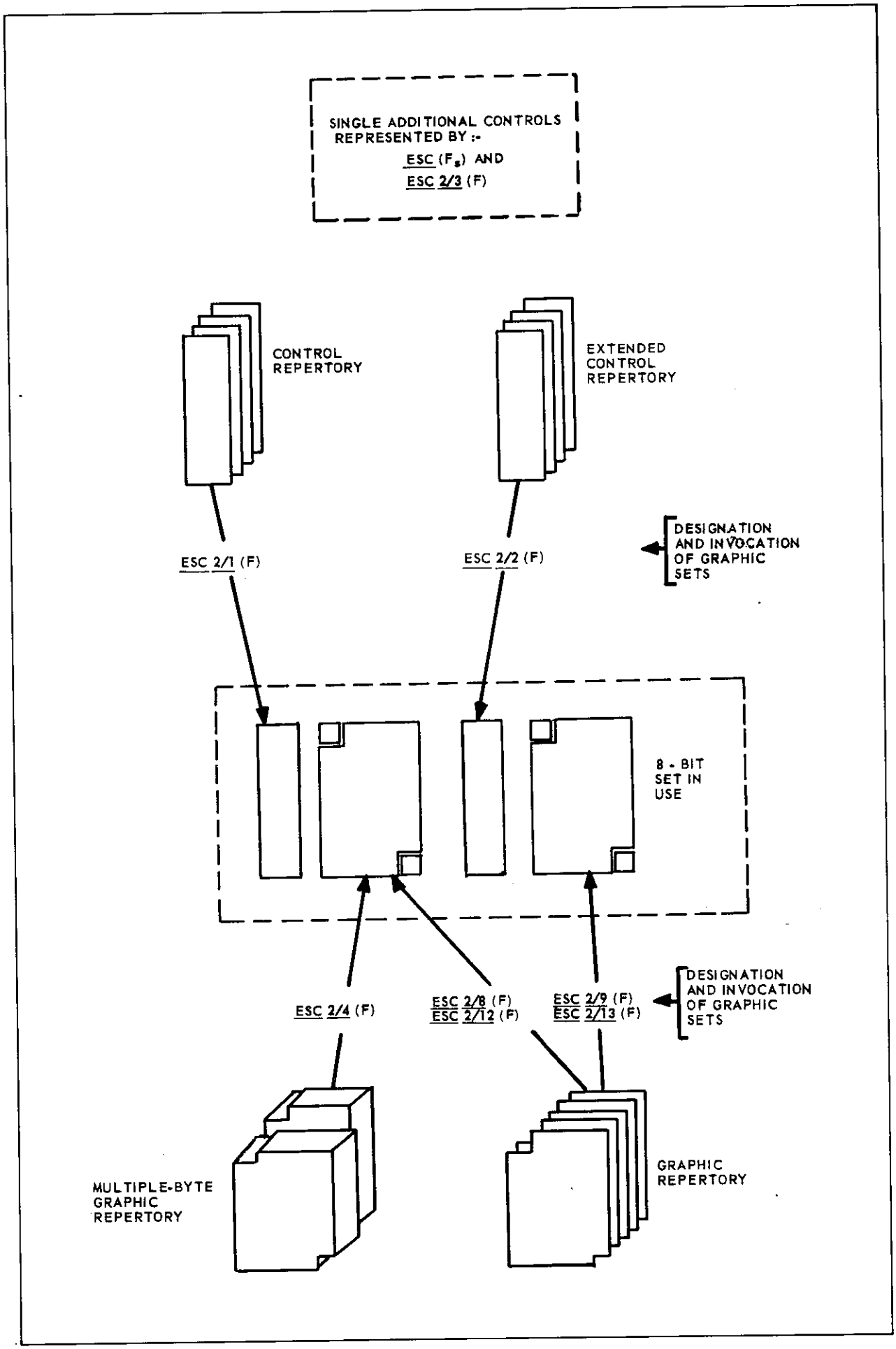


Fig. 9.2 Code extension in an 8-bit environment

Table 9.1 International standardized codes and the Swedish codes derived from them

International Standardized Codes

6 bits ISO R/646 entitled "6 and 7 Bit Coded Character Sets for Information Interchange". 1st edition, Dec. 1967.

The 6-bit code has been omitted from the new 2nd edition appearing in ISO/TC97/SC2 N566 dated Oct. 1971.

7 bits ISO R/646 entitled "7 Bit Coded Character Set for Information Interchange". New 2nd edition in ISO/TC97/SC2 N566 dated Oct. 1971.

Standard ECMA-6 entitled "7 Bit Input/Output Coded Character Set". 3rd edition, July 1970.

8 bits ISO/TC97/SC2 N562 entitled "Code Extension Techniques for Use with the 7-Bit Coded Character Set of ISO R/646" dated Oct. 1971.

Standard ECMA-35 entitled "Extension of the 7 Bit Coded Character Set". Dec. 1971.

Swedish Standard Codes Specified in Swedish Electrotechnical Standard 850200

This standard presents codes that are based on the above international codes and contain the following characters. It is expected that this standard will be formally adopted during 1972.

	Position	Character	
6 bits	3/11	Ä	
	3/12	Ö	
	3/13	Å	
7 bits	2/3	#	Version 1
	2/4	¤	
	5/11	Ä	
	5/12	Ö	
	5/13	Å	
	7/11	ä	Version 2
	7/12	ö	
	7/13	å	
	4/0	É	
	5/14	Ü	
	6/0	é	
	7/14	ü	