UNITED NATIONS INDUSTRIAL DEVELOPMENT ORGANIZATION
Vienna International Centre, P.O. Box 300, 1400 Vienna, Austria
Tel: (+43-1) 26026-0 · www.unido.org · unido@unido.org

TÜBİTAK A.E.A.G.E.

Hardware Modification of Training Simulator in Kütahya Training Center

Final Report

Assoc.Prof.Dr.İsmet ERKMEN

M.Sedat UĞUR

February 25th, 1991

INTRODUCTION

The aim of this project is to develop and run the following two software programs to be used with SIMTRAN simulator after the necessary hardware modifications have been performed.

- •- Dilute nitric acid production
- Ammonia nitrade production

The existing SIMTRAN simulator being custom designed is not accessible for software development. For this reason the computer section of the simulator is planned to be replaced by an IBM-PC compatible computer with the necessary interfacing unit. This work is carried out by TÜBİTAK Ankara Electronics Research and Development section.

The software programs will be developed on the basis of the actual processes used at the Kütahya plant. This work is carried out by TÜBİTAK Marmara Scientific and Industrial Research Institute, Chemical Engineering Department.

The processes achived by the above mentioned Institutes are described in the following sections.

WORK PERFORMED

The hardware modification of the existing SIMTRAN simulator requires the upgrading of the custom designed computer with an IBM-PC compatible computer. For this purpose the following interfacing unit have been designed and implemented.

During the tests and studies on the existing SIMTRAN simulator two design mistakes, which were causing malfunctioning of origional system under certain conditions, have been noted and modified for the proper operation as detailed in the report.

Parallel Interface Unit

The task of the unit which will plug into the I/O bus of the computer is to take the data from the computer and to send it to the digital circuitry of the student's console. The unit can be divided into two main sections, the parallel interface and the line driver sections.

i.) Parallel interface section: This section consists of Intel 8255 Programmable Parallel Adapter and the necessary address decoding circuitry. The circuit diagram is given at the Appendices as drawing No.2.

The addressing of the Intel 8255 which will be used by the simulation software is as follows;

Port A	:	Hex 300
Port B	:	Hex 301
Port C	:	Hex 302
Control	:	Hex 303

Eight bits of data is sent or received via Port A (PA) and three bits are send to control the operation via Port B of 8255. Two bits of them (PB2, PB3) are dedicated to indicate the direction of data as given below;

PB3	PB2	
1	0	; writestr=1, data from PC to Student's console
0	1	; readstr=1, data from Student's console to PC
1	1	; PC and the Student's console are disconnected.

The third one of the control bits (PB4, ALE=Address Latch Enable) is a probe signal used to latch the address sent from the computer to the interface circuitry of the student's console.

There is a simple protocol that must be obeyed to read/write data from/to the devices (indicators, controllers, etc.) on the Student's console. First step is to write the address of the device to Port A. Then with a high to low transition of ALE the address is latched in and decoded by the interface circuitry of the Student's console. Finaly data can be read/written from/to the selected device.

Below is a routine which is written in PASCAL which implements the protocol to write data to the console;

```
port[303h]:=80h          ; Initialize 8255
                         ; Set all the ports as output
port[301h]:=1Ch          ; isolate the drivers
port[300h]:=address      ; write the device address to Port A
port[301h]:=18h          ; ALE=High & drivers=Outwards
port[301h]:=08h          ; ALE=Low, ie. latch address to buffers
                         ; of the Student's Console
port[300h]:=data         ; write data to Port A
port[301h]:=1Ch          ; ALE=High & isolate the drivers
```

The routine which reads data from the console is as follows;

```
port[303h]:=80h          ; Initialization of 8255
                         ; Set all the ports as output
port[301h]:=1Ch          ; isolate the drivers
port[300h]:=address      ; write the device address to Port A
port[301h]:=18h          ; ALE=High drivers : outward
port[301h]:=08h          ; ALE=Low, latch the address to buffers
port[301h]:=18h          ; bring ALE to high
port[303h]:=90h          ; PortA input
port[301h]:=14h          ; ALE=High  drivers:inwards
data:=port[300h]         ; Read portA
port[301h]:=1Ch          ; Bring drivers back to
                         ; isolated position
```

where 'address' is the address of a device on the student's console and 'data' is the received or transmitted data. The whole listing of the addresses is given in the appendices.

ii.) Line Driver Section: The task of this circuit is to compensate the noise added to the signals due to the long cable between the Student's and Instructor's Consoles. The signals relative to ground are isolated from the ground and transmitted differentialy for each bit. The circuit diagram is given as Drawing No.1 in the Appendices.

Modifications Made On The Student's Console:

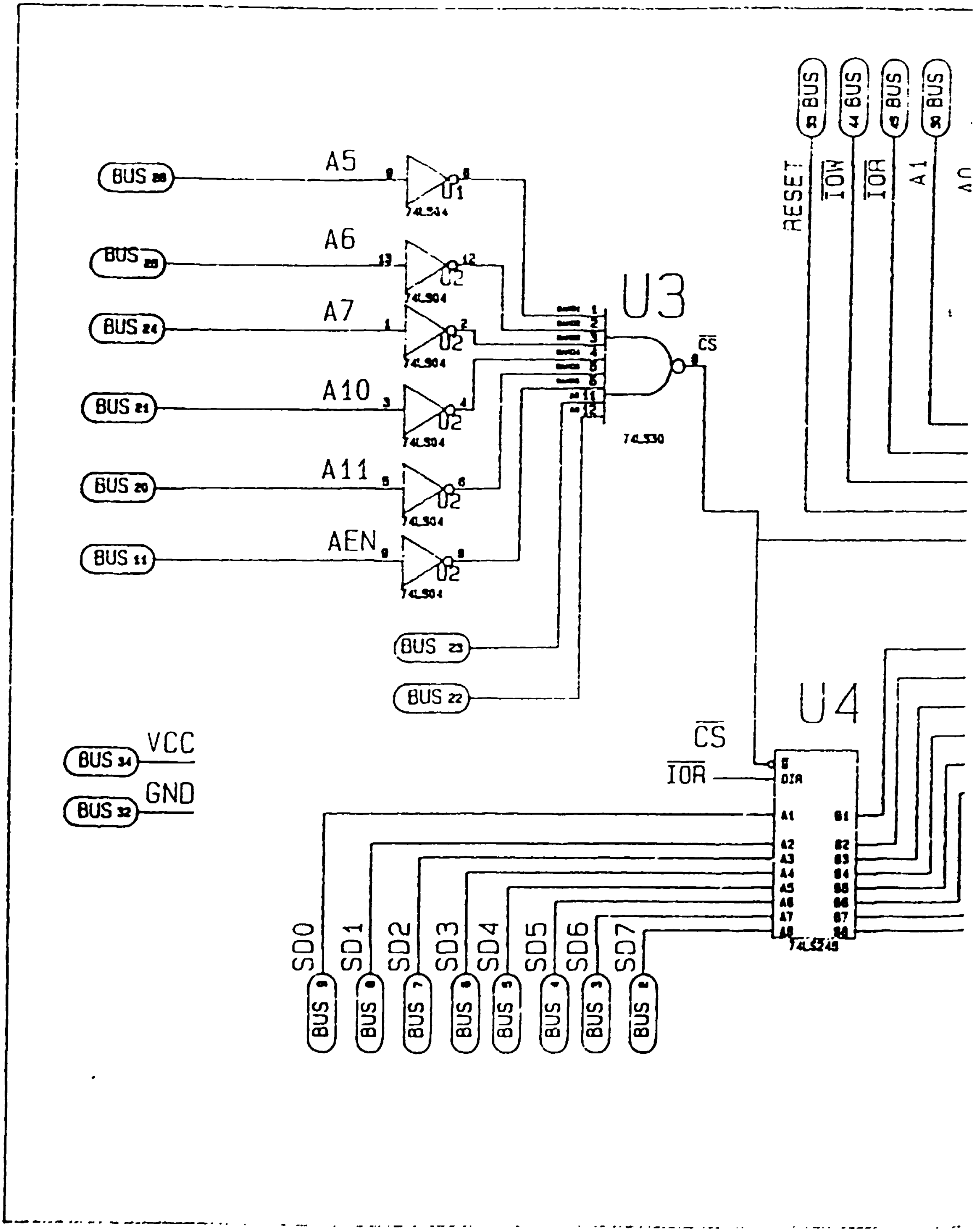
There are twelve Foxboro brand PID (proportional, integral, derivative)controllers on the student console. These devices generate an output voltage to control a preassigned parameter around a pre-set value. Deviation of the parameter from the pre-set value is called the error signal. Output voltage is proportional to the present value, rate of change and accumulation of this error signal so as to keep the parameter as close as to the pre-set value. Parameter is fictitious and it is generated by the computer software. These parameters are fed to the controllers through digital circuitry. Unfortunately there was an inherent potential difference between the analog ground (which feeds controllers) and the digital ground (which feeds the digital circuits) causing a lot of trouble. Input parameters sent to the controllers from the digital circuitry together with the inherent potential difference added on them were out of the linear operation region of the controllers. Therefore even with a small input voltage the controllers became saturated. This problem could be solved if somehow one could remove the potential difference between the digital and analog grounds. An extreme solution was to short circuit these grounds zero-ing the difference, however for convenience a 68 Ohms resistor was installed in between causing an excess current of 6-7mA which can be considered as minute. Installation of this 68 Ohms resistor pulled the controllers into the linear operation region.

Student console is also propped up with an annunciator which is nothing but a buzzer and several lamps blinking used to warn the user of some parameters exceeding or falling below a pre-determined value. On our tests to access this annunciator using our new computer it was observed that even they were not triggered lamps on the annunciator began blinking in a random fashion causing erroneous warning. Carrying out a deep search on the student console circuitry the source of this random behaviour was determined to be the noise along the annunciator supply. Noise was due to the annunciator buzzer which was a mechanical ring generating clicks along the supply path. Surely supply path was coupling these clicks to the annunciator circuitry. This noise was greatly suppressed when a low-pass filter was installed on the buzzer, softening the clicks. Consequently accesses from the new computer proved a proper annunciator operation.

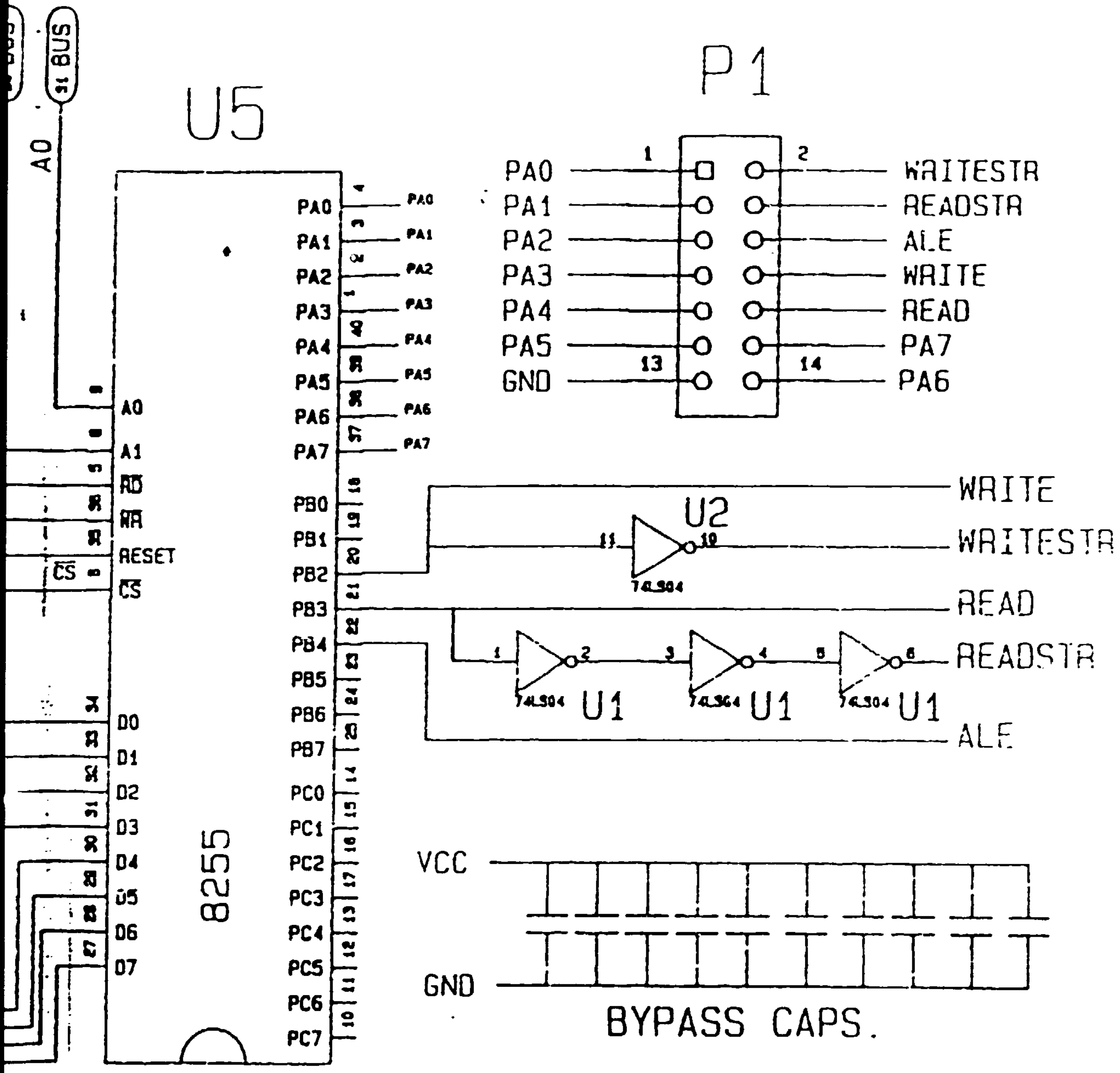
APPENDIX

1. Circuit diagram of the Parallel Interface Card

2. Process Control Simulator I/O structure

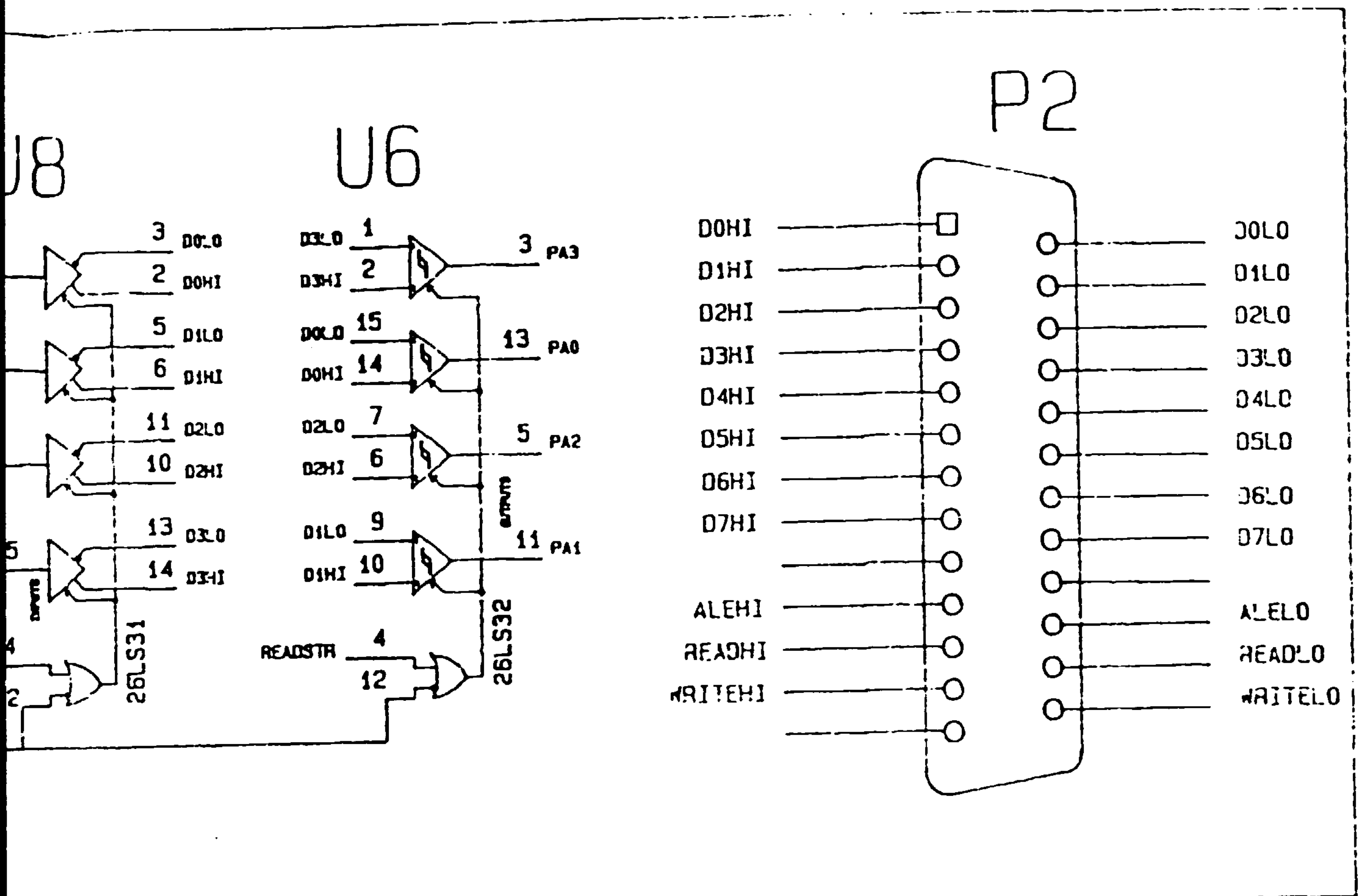


BUS 26
A5
U1
74LS04
BUS 25
A6
U2
74LS04
BUS 24
A7
U2
74LS04
BUS 21
A10
U2
74LS04
BUS 20
A11
U2
74LS04
BUS 11
AEN
U2
74LS04
BUS 23
BUS 22
U3
CS
74LS30
RESET
IOW
IOR
A1
A0
BUS
BUS
BUS
BUS
BUS 34
VCC
BUS 32
GND
U4
CS
IOR
G
DIR
A1
A2
A3
A4
A5
A6
A7
A8
B1
B2
B3
B4
B5
B6
B7
B8
74LS245
SD0
SD1
SD2
SD3
SD4
SD5
SD6
SD7
BUS 9
BUS 8
BUS 7
BUS 6
BUS 5
BUS 4
BUS 3
BUS 2



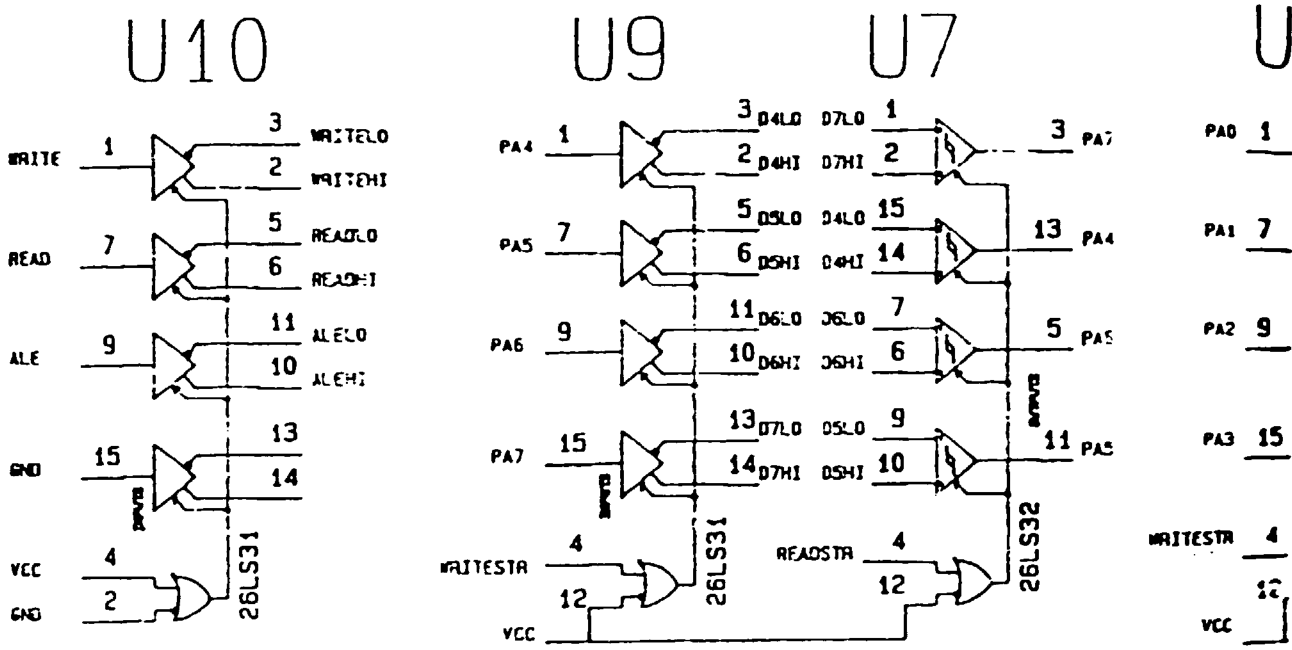
TUBITAK
AFAGE

KUTAHYA - AZOT
PARALLEL INTERFACE
MARCH 13, 1989 DRW #2



TUBITAK
AEAGE

KUTAHYA AZOT
PARALLEL INTERFACE
MARCH 13, 1989
DRW #1



COMPONENT LIST

U6, U7	DS26LS32	P1	25 PIN CONNECTOR
U10, U9, U8	DS26LS31		
U5	MSM8255A		
U1, U2	74LS04	P2	14 PIN CONNECTOR
U4	DM74LS245		

PROCESS CONTROL SIMULATOR I/O STRUCTURE

OVERVIEW

		(HEXADECIMAL EQUIVALENT)
ADRØ - ADR15	SWITCH INPUTS	(ØØ-ØF)
ADR16 - ADR31	ANALOG INPUTS	(1Ø-1F)
ADR32 - ADR47	SPARE BLOCK	(2Ø-2F)
ADR48 - ADR63	DVM OUTPUTS	(3Ø-3F)
ADR64 - ADR79	METER OUTPUTS	(4Ø-4F)
ADR80 - ADR95	METER OUTPUTS	(5Ø-5F)
ADR96 - ADR111	SPARE	(6Ø-6F)
ADR112 - ADR127	SPARE	(7Ø-7F)

SWITCH INPUTS

ADRØ	(HEX ØØ)	1 2 4 8	RECORDER 1 (Red Pen)
		1 2 4 8	RECORDER 2 (Red Pen)
ADR1	(HEX Ø1)	1 2 4 8	RECORDER 3 (Red Pen)
		1 2 4 8	RECORDER 4 (Red Pen)
ADR2	(HEX Ø2)	1 2 4 8	DVM1
		1 2 4 8	DVM2
ADR3	(HEX Ø3)	PUMP 1 PUMP 2 PUMP 3 PUMP 4 SPARE SPARE SPARE	
ADR4	(HEX Ø4)	1 2 4 8	RECORDER 1 (Blue Pen)
		1 2 4 8	RECORDER 2 (Blue Pen)
ADR5	(HEX Ø5)	1 2 4 8	RECORDER 3 (Blue Pen)
		1 2 4 8	RECORDER 4 (Blue Pen)
ADR6	(HEX Ø6)	1 2 4 8	RECORDER 1 (Green Pen)
		1 2 4 8	RECORDER 2 (Green Pen)
ADR7	(HEX Ø7)	1 2 4 8	RECORDER 3 (Green Pen)
		1 2 4 8	RECORDER 4 (Green Pen)
ADR8 - ADR15	(HEX Ø8 - ØF)		SPARE

ANALOG INPUTS

ADR16	CONTROLLER 1	(HEX1Ø)
ADR17	CONTROLLER 2	(HEX11)
ADR18	CONTROLLER 3	(HEX12)
ADR19	CONTROLLER 4	(HEX13)
ADR20	CONTROLLER 5	(HEX14)
ADR21	CONTROLLER 6	(HEX15)
ADR22	CONTROLLER 7	(HEX16)
ADR23	CONTROLLER 8	(HEX17)
ADR24	CONTROLLER 9	(HEX18)
ADR25	CONTROLLER 10	(HEX19)
ADR26	CONTROLLER 11	(HEX1A)
ADR27	CONTROLLER 12	(HEX1B)
ADR28	VALVE 1	(HEX1C)
ADR29	VALVE 2	(HEX1D)
ADR30	VALVE 3	(HEX1E)
ADR31	VALVE 4	(HEX1F)

SPARE BLOCK

ADR32 - ADR47	SPARES	(HEX2Ø - 2F)

DVM OUTPUTS

ADR48	DVM 1-10K BIT
(HEX 30)	20K BIT
	40K BIT
	80K BIT
	DP2
	DP3
	DP4
	BLANKING
ADR49	DVM 1-100 BIT
(HEX 31)	200 BIT
	400 BIT
	800 BIT
	1K BIT
	2K BIT
	4K BIT
	8K BIT
ADR50	DVM 1-1 BIT
(HEX 32)	2 BIT
	4 BIT
	8 BIT
	10 BIT
	20 BIT
	40 BIT
	80 BIT
ADR51	DVM 2-10K BIT
(HEX 33)	20K BIT
	40K BIT
	80K BIT
	DP 2
	DP 3
	DP 4
	BLANKING
ADR52	DVM 2-100 BIT
(HEX 34)	200 BIT
	400 BIT
	800 BIT
	1K BIT
	2K BIT
	4K BIT
	8K BIT
ADR53	DVM 2-1 BIT
(HEX 35)	2 BIT
	4 BIT
	8 BIT
	10 BIT
	20 BIT
	40 BIT
	80 BIT
ADR54	SPARE
(HEX 36)	

ADR55	ANNUNCIATOR - CH 1
(HEX 37)	ANNUNCIATOR - CH 2
	ANNUNCIATOR - CH 3
	ANNUNCIATOR - CH 4
	ANNUNCIATOR - CH 5
	ANNUNCIATOR - CH 6
	ANNUNCIATOR - CH 7
	ANNUNCIATOR - CH 8
ADR56	ANNUNCIATOR - CH 9
(HEX 38)	ANNUNCIATOR - CH 10
	ANNUNCIATOR - CH 11
	ANNUNCIATOR - CH 12
	ANNUNCIATOR - CH 13
	ANNUNCIATOR - CH 14
	ANNUNCIATOR - CH 15
	ANNUNCIATOR - CH 16
ADR57	PUMP 1 LAMP
(HEX 39)	PUMP 2 LAMP
	PUMP 3 LAMP
	PUMP 4 LAMP
	SPARE
	SPARE
	SPARE
	SPARE
ADR58 - ADR63	SPARES
(HEX3A - 3F)	

METER OUTPUTS

ADR64	CONTROLLER 1	(HEX 4Ø)
ADR65	CONTROLLER 2	(HEX 41)
ADR66	CONTROLLER 3	(HEX 42)
ADR67	CONTROLLER 4	(HEX 43)
ADR68	CONTROLLER 5	(HEX 44)
ADR69	CONTROLLER 6	(HEX 45)
ADR70	CONTROLLER 7	(HEX 46)
ADR71	CONTROLLER 8	(HEX 47)
ADR72	CONTROLLER 9	(HEX 48)
ADR73	CONTROLLER 10	(HEX 49)
ADR74	CONTROLLER 11	(HEX 4A)
ADR75	CONTROLLER 12	(HEX 4B)
ADR76	RECORDER 1 (RED)	(HEX 4C)
ADR77	RECORDER 2 (RED)	(HEX 4D)
ADR78	RECORDER 3 (RED)	(HEX 4E)
ADR79	RECORDER 4 (RED)	(HEX 4F)
ADR80	RECORDER 1 (BLUE)	(HEX 5Ø)
ADR81	RECORDER 2 (BLUE)	(HEX 51)
ADR82	RECORDER 3 (BLUE)	(HEX 52)
ADR83	RECORDER 4 (BLUE)	(HEX 53)
ADR84	RECORDER 1 (GREEN)	(HEX 54)
ADR85	RECORDER 2 (GREEN)	(HEX 55)
ADR86	RECORDER 3 (GREEN)	(HEX 56)
ADR87	RECORDER 4 (GREEN)	(HEX 57)
ADR88	INDICATOR 1	(HEX 58)
ADR89	INDICATOR 2	(HEX 59)
ADR90	INDICATOR 3	(HEX 5A)
ADR91	INDICATOR 4	(HEX 5B)
ADR92	INDICATOR 5	(HEX 5C)
ADR93	INDICATOR 6	(HEX 5D)
ADR94-ADR95	SPARE	(HEX 5E-5F)
ADR96-ADR111	SPARE	(HEX 6Ø-6F)
ADR112-ADR127	SPARE	(HEX 7Ø-7F)