ADVANCE INFORMATION

TPU 3041
Cyrillic/Greek
Teletext Processor

ITT Semiconductors ITT

Contents

Multistandard Teletext Processor for Level 1 and 2

> **Release Notes:**
>
> **This paper describes only those functions and characteristics of the TPU 3041 which are different to functions and characteristics of the TPU 3040.**
>
> **For detailed information on common features please read the TPU 3040 data sheet.**
>
> **The information given applies to TPU 3041, technical code 22.**
> **Revision bars indicate significant changes to the previous version.**

1. Introduction

The TPU3041 is a single-chip World System Teletext (WST) decoder for applications in analog and digital TV sets. **It is designed for use in countries with Greek or Cyrillic character sets. The Latin character set is still supported including a wide range of national characters.**

1.1. Technical Code Change from TC19 to TC22

TPU3041 TC22 is based on hardware and firmware of TPU3040 TC20. The following table shows the improvements compared to TC18.

TPU 3040	TC10 TC12	TC13	TC14 TC16	TC17 TC18	TC20
Display					
Stop rolling time/header in parallel magazin mode	x	x	x	x	
Different italics timing in 10MHz analog mode Problem: Italics characters are displayed by shifting pixel in steps of 1/4 clock periods. In 10 MHz analog mode, pixels are shifted by 3/4 clock periods instead.				x	
Row 26 character flicker without parity error	x	x	x	x	
Row 26 overwriting pointers in row0 Problem: row 26 can address columns 32–39 in row 0 Solution: set bit "row 26 hold" in display mode				x	
Row 26 not checking row 24 hold	x	x	x	x	
Double height flicker after page clear	x	x	x	x	
No support of Serbian language in character set 38	x	x	x	x	
PRIO Interface					
Jitter in MSYNC interface Problem: display jitter of 1 pixel Solution: enable skew delay	x	x	x	x	
Display flicker during channel search	x	x	x	x	
Memory Manager					
Ghost rows overwriting memory Problem: On channels with parallel magazine transmission, the memory manager of TPU3040 can destroy ghost row pointers. Received ghost rows are then written randomly into memory. Solution: 4 Mbit DRAM				x	

TC22 also includes some firmware improvements compared to TC20. Please check TPU3041 data sheet for detailed description of the following commands:

Command 07   "Acquisition Mode"
Command 18   "Read Reset Source"
Command 26   "Display Page Request"

1.2.   Software Incompatibility between Technical Codes 19 and 22

TPU3041, Technical Code 19, did not include the user font of TPU3040. The user font is used in some applications to generate OSD menus. To maintain compatibility with existing TPU3040 applications and to guarantee compatibilty to future versions, such as TPU3042 (Arabic/Hebrew), it was decided to change the font organization of TPU3041 from TC19 to TC22. This means that the character mapping in the Cyrillic font is changed. **External software generating OSD texts must use new mapping tables.**

In TC 22, command 60 was improved and now supports all features of the full row attribute, such as double height. Therefore it is not compatible to command 60 in TC19. Please check TPU3041 data sheet for a more detailed description.

2. Functional Description

The original G2 font and the NTSC fonts of TPU3040 had to be removed from the character ROM.

The firmware was modified to support the additional fonts "Greek" and "Cyrillic" using ESC code (1/11) to switch between different character sets inside a teletext page. The ESC code is only used during translation of a teletext page from page memory into TTX display bank. In the display bank the ESC code has no function.

Addressing of the fonts "Greek" and "Cyrillic" is achieved by inserting level 2 parallel attributes into the display bank. The existence of level 2 data in the TTX display bank is a major difference between TPU3040 and TPU3041 and made some additional commands necessary. Command 57 "*Display Read Column*" can be used to read the full row attributes from the display bank. Command 60 "*Display Full Row*" can be used to set the full row attributes of the display bank without resetting the level 2 bit.

The two character sets which are exchanged by ESC code can be defined with additional commands not available in TPU3040. These commands (58 and 59) can be used to define two character sets including eight languages each. Table 2–1 shows the available languages in TPU3041.

Table 2–1: Language table

Code	Language
0	English
1	French
2	Swedish, Finnish
3	Czech
4	German
5	Spanish
6	Italian
7	Estonian, Finnish
8	English (US)
9	Slovakian
10	Hungarian
11	Serbian, Croatian, Slovene
12	Albanian
13	Polish
14	Turkish
15	Rumanian
16	Cyrillic (Russian, Bulgarian)
17	Greek
18	Cyrillic (Serbian, Montenegro)
19	YU Latin (new)
20–255	not defined

3. Definitions

3.1. Character Set

10-bit Character Code

000H	////////////
	G0
	G0
	G0
080H	National
	National
	National
	National

100H	////////////
	G1
	G0
	G1
180H	National
	National
	National
	National

200H	////////////
	G2
	G2
	G2
280H	User
	User
	User
	User

300H	////////////
	G3
	G3
	G3
380H	User
	User
	User
	User

Mapping ROM
32 x 5 bit

Character ROM

G0
G0
G0
G1
G1
User
National
National
National
National
Greek
Greek
Greek
Cyrillic
Cyrillic
Cyrillic

32 char = 400 byte
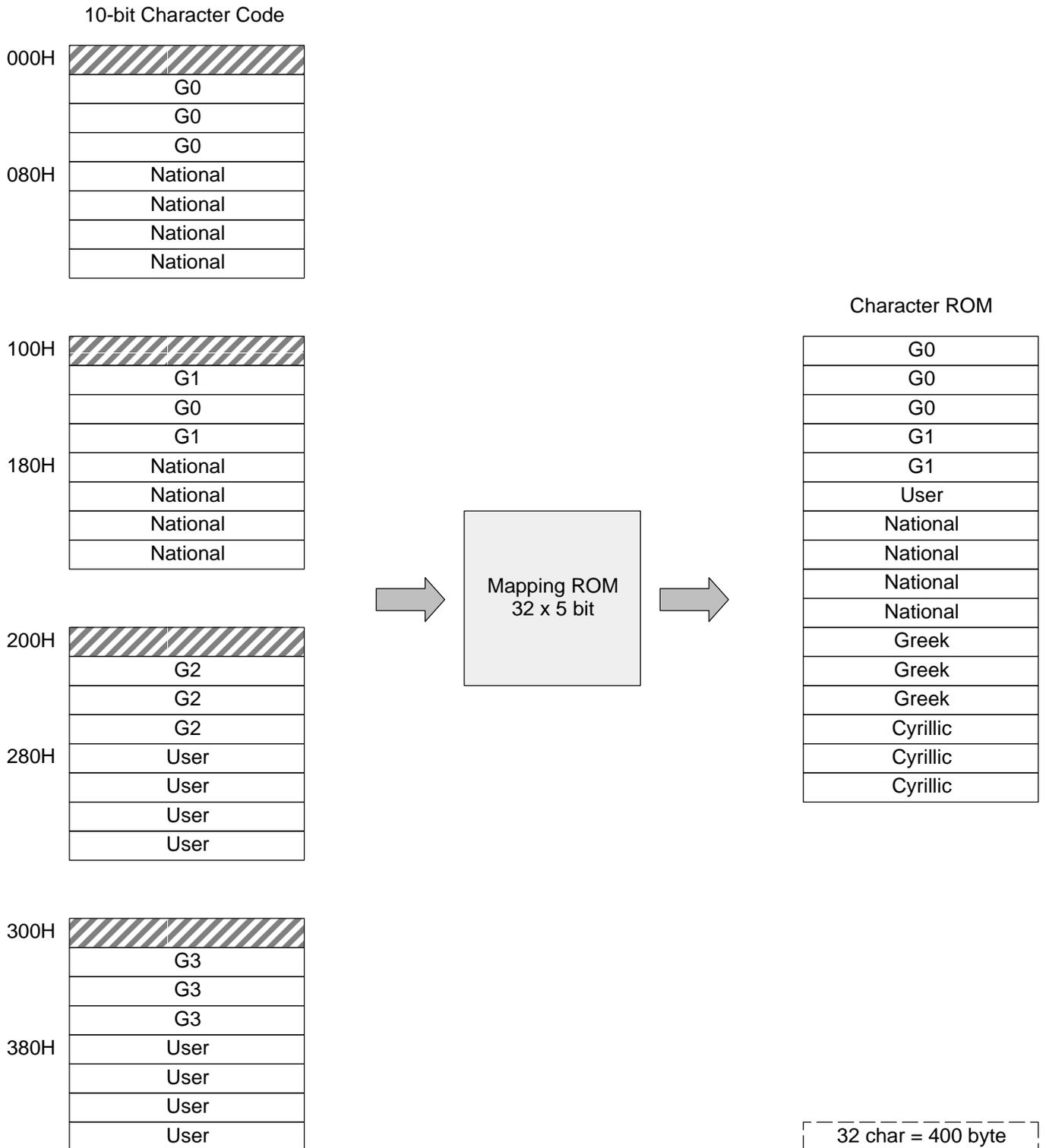
Fig. 3–1: Character Set Organization

3.2. Character Font

Table 3–1: G0 font

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0																
1																
2		!	"	£	$	%	&	'	(	)	*	+	,	–	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	←	½	→	↑	#
6	—	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	¼	‖	¾	÷	■

◸ = National Option

Table 3–2: National font

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
8	Ç	ï	ù	á	í	à	ì	â	î	û	ä	ö	ü	Ä	Ö	Ü
9	ç	ë	ú	é	ó	è	ò	ê	ô	¿	ß	ñ	å	Å	É	¡
10	Ï	Á	Í	À	Ì	Â	Î	Ã	Ĩ	ę	Ÿ	ť	ğ	ğ	Ţ	ţ
11	É	Ę	Ó	È	Ò	Ê	Ô	ã	ð	ă	õ	õ	Š	š	Ő	ő
12	ě	Ě	Ú	Đ	Ù	ă	Û	ś	ŏ	ő	Ÿ	ý	ž	Ž	Ű	ű
13	_	đ	ć	ć	·	\	π	ś	ż	ż	ź	ž	Č	Č	ę	ə
14	º	Ħ	Ń	ń	Ľ	ↄ	^	`	Ľ	ľ	Ñ	ň	Ň	ň	Ş	ş
15	Ş	ı	Ŕ	ŕ	⊐	⊃			˜	ŧ	ŧ	Ļ	Ľ	Ŗ	ř	ə	º

Table 3–3: G1 font

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0																
1																
2																
3																
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	←	½	→	↑	#
6																
7																

▽ = National Option

Table 3–4: G2 font

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0																
1																
2		!	ʼʼ	#	$	%	&	'	⟨	⟩	✳	+	,	−	.	∕
3	0	1	2	3	4	5	6	7	8	9	:	;	≪	=	≫	?
4	Ϊ	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο
5	Π	Ρ	_	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Ϊ	Ϋ	ά	έ	ή	ί
6	ϋ	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο
7	π	ρ	ς	σ	τ	υ	φ	χ	ψ	ω	ϊ	ϋ	ό	ύ	ώ	■

Table 3–5: User font

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
8	▲	▼	⌐	⌐	—	Ɛ	Ɛ	│	├	⌐	_	▬	‾	▶	◀	©
9	└	┘	┐	┌	_	│	‾	·	·	.	.	☰	☰	△	▽	☺
10	Б	Г	Д	Ђ	Ж	З	И	Л	Љ	Њ	П	Ћ	У	Ф	Ц	Ч
11	Џ	Ш	Љ	Њ	Dž	Ю	Й	Я	Ь	Ъ	Щ	Ы	Ќ	Ѓ	Э	Є
12	б	г	д	ђ	ж	з	и	л	љ	њ	п	ћ	у	ф	ц	ч
13	џ	ш	lj	nj	dž	ю	й	я	ь	ъ	щ	ы	ќ	ѓ	э	є
14	м	н	т	в	Ў	ў	Ґ	ґ	к	І	і	Ї	ї	°	Ҍ	Ѵ
15	Ώ	Ά	Ή	Έ	—	├	┼	+	┿	┤	_	└	▙	▟	■	■

3.3. Latin Character Mapping

Table 3–6: Character set options

Option Bits	Character Set					
C14,C13,C12	6	38	40	55	70	128
000	English	Polish	English (US)	English	English (US)	programmable
001	French	French	French	French	Slovakian	programmable
010	Swedish	Swedish	Swedish	Swedish	Hungarian	programmable
011	Czech	Czech	Czech	Turkish	Serbian	programmable
100	German	German	German	German	Albanian	programmable
101	Spanish	Serbian	Spanish	Spanish	Polish	programmable
110	Italian	Italian	Italian	Italian	Turkish	programmable
111	Estonian	Estonian	Estonian	Estonian	Rumanian	programmable

Table 3–7: National option mapping

Language	G0/G1 Table Position												
	2/3	2/4	4/0	5/11	5/12	5/13	5/14	5/15	6/0	7/11	7/12	7/13	7/14
Albanian	5/15	2/4	13/12	13/2	12/12	12/3	11/12	9/1	13/13	13/3	12/13	13/1	11/13
Czech	5/15	12/9	13/13	10/11	12/13	12/11	8/4	15/13	9/3	8/3	12/0	9/2	11/13
English	2/3	2/4	4/0	5/11	5/12	5/13	5/14	5/15	6/0	7/11	7/12	7/13	7/14
English (US)	5/15	2/4	4/0	14/4	13/5	15/4	14/6	13/0	14/7	14/5	15/6	15/5	15/7
Estonian, Finnish	5/15	11/11	11/12	8/13	8/14	12/12	8/15	11/10	11/13	8/10	8/11	12/13	8/12
French	9/3	8/1	8/5	9/1	9/7	8/2	8/8	5/15	9/5	8/7	9/8	8/9	9/0
German	5/15	2/4	15/0	8/13	8/14	8/15	14/6	13/0	14/0	8/10	8/11	8/12	9/10
Hungarian	5/15	9/2	9/14	8/4	8/14	10/1	12/15	11/15	9/3	9/4	8/11	8/3	8/12
Italian	2/3	2/4	9/3	14/0	9/0	5/13	5/14	5/15	8/2	8/5	9/6	9/5	8/6
Polish	5/15	14/3	13/15	13/8	12/7	15/8	13/3	9/4	10/9	13/9	13/7	15/9	13/11
Rumanian	5/15	14/1	10/14	10/5	14/14	14/11	10/6	15/1	10/15	8/7	14/15	12/5	8/8
Serbian, Croatian	5/15	2/4	13/12	13/2	12/12	12/3	11/12	13/0	13/13	13/3	12/13	13/1	11/13
Slovakian	5/15	12/9	13/13	10/11	12/13	12/11	8/4	15/13	9/3	8/3	12/0	9/2	11/13
Spanish	9/0	2/4	9/15	8/3	9/3	8/4	9/4	9/2	9/9	8/12	9/11	9/5	8/5
Swedish, Finnish	5/15	14/1	9/14	8/13	8/14	9/13	8/15	13/0	9/3	8/10	8/11	9/12	8/12
Turkish	13/6	10/13	10/8	14/14	8/14	8/0	8/15	10/12	15/1	14/15	8/11	9/0	8/12
YU Latin	5/15	2/4	13/12	13/2	12/12	12/3	11/12	13/0	13/13	13/3	12/13	13/1	11/13

3.4. Cyrillic Character Mapping

The Cyrillic font definition of TPU3041 does not match the official specifications of the Cyrillic G0 character set exactly as national options and packet 26 characters are included.

The TPU3041 firmware uses tables to map the received character code into the internal font definition. The same tables should be used by external software for generation of Cyrillic OSD text.

Table 3–8: Cyrillic G0 mapping

```
char G_RUSSIAN_CYRILLIC_LEVEL_1[] = {
    0x20, 0x21, 0x22, 0x5f, 0x24, 0x25, 0xdb, 0x27,
    0x28, 0x29, 0x2a, 0x2b, 0x2c, 0x2d, 0x2e, 0x2f,
    0x30, 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37,
    0x38, 0x39, 0x3a, 0x3b, 0x3c, 0x3d, 0x3e, 0x3f,
    0xb5, 0x41, 0xa0, 0xae, 0xa2, 0x45, 0xad, 0xa1,
    0x58, 0xa6, 0xb6, 0x4b, 0xa7, 0x4d, 0x48, 0x4f,
    0xaa, 0xb7, 0x50, 0x43, 0x54, 0xac, 0xa4, 0x42,
    0xb8, 0xb9, 0xa5, 0xb1, 0xbe, 0xba, 0xaf, 0xbb,
    0xd5, 0x61, 0xc0, 0xce, 0xc2, 0x65, 0xcd, 0xc1,
    0x78, 0xc6, 0xd6, 0xe8, 0xc7, 0xe0, 0xe1, 0x6f,
    0xca, 0xd7, 0x70, 0x63, 0xe2, 0xcc, 0xc4, 0xe3,
    0xd8, 0xd9, 0xc5, 0xd1, 0xde, 0xda, 0xcf, 0x7f
    };
char G_RUSSIAN_CYRILLIC_LEVEL_2[] = {
    0x60, 0x60, 0x60, 0x60, 0x60, 0x60, 0x62, 0x60,
    0x60, 0x60, 0x60, 0x60, 0x60, 0x60, 0x60, 0x60,
    0x60, 0x60, 0x60, 0x60, 0x60, 0x60, 0x60, 0x60,
    0x60, 0x60, 0x60, 0x60, 0x60, 0x60, 0x60, 0x60,
    0x62, 0x60, 0x62, 0x62, 0x62, 0x60, 0x62, 0x62,
    0x60, 0x62, 0x62, 0x60, 0x62, 0x60, 0x60, 0x60,
    0x62, 0x62, 0x60, 0x60, 0x60, 0x62, 0x62, 0x60,
    0x62, 0x62, 0x62, 0x62, 0x62, 0x62, 0x62, 0x62,
    0x62, 0x60, 0x62, 0x62, 0x62, 0x60, 0x62, 0x62,
    0x60, 0x62, 0x62, 0x62, 0x62, 0x62, 0x62, 0x60,
    0x62, 0x62, 0x60, 0x60, 0x62, 0x62, 0x62, 0x62,
    0x62, 0x62, 0x62, 0x62, 0x62, 0x62, 0x62, 0x60
    };
char G_SERBIAN_CYRILLIC_LEVEL_1[] = {
    0x20, 0x21, 0x22, 0x5f, 0x24, 0x25, 0x26, 0x27,
    0x28, 0x29, 0x2a, 0x2b, 0x2c, 0x2d, 0x2e, 0x2f,
    0x30, 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37,
    0x38, 0x39, 0x3a, 0x3b, 0x3c, 0x3d, 0x3e, 0x3f,
    0xaf, 0x41, 0xa0, 0xae, 0xa2, 0x45, 0xad, 0xa1,
    0x58, 0xa6, 0x4a, 0x4b, 0xa7, 0x4d, 0x48, 0x4f,
    0xaa, 0x51, 0x50, 0x43, 0x54, 0xac, 0x42, 0x57,
    0x58, 0x59, 0xa5, 0xab, 0xa4, 0xa3, 0xb1, 0xd0,
    0xcf, 0x61, 0xc0, 0xce, 0xc2, 0x65, 0xcd, 0xc1,
    0x78, 0xc6, 0x6a, 0xe8, 0xc7, 0xe0, 0xe1, 0x6f,
    0xca, 0x71, 0x70, 0x63, 0xe2, 0xcc, 0xe3, 0x77,
    0x78, 0x79, 0xc5, 0xcb, 0xc4, 0xc3, 0xd1, 0x7f
    };
char G_SERBIAN_CYRILLIC_LEVEL_2[] = {
    0x60, 0x60, 0x60, 0x60, 0x60, 0x60, 0x60, 0x60,
    0x60, 0x60, 0x60, 0x60, 0x60, 0x60, 0x60, 0x60,
    0x60, 0x60, 0x60, 0x60, 0x60, 0x60, 0x60, 0x60,
    0x60, 0x60, 0x60, 0x60, 0x60, 0x60, 0x60, 0x60,
    0x62, 0x60, 0x62, 0x62, 0x62, 0x60, 0x62, 0x62,
    0x60, 0x62, 0x60, 0x60, 0x62, 0x60, 0x60, 0x60,
    0x62, 0x60, 0x60, 0x60, 0x60, 0x62, 0x60, 0x60,
    0x60, 0x60, 0x62, 0x62, 0x62, 0x62, 0x62, 0x60,
    0x62, 0x60, 0x62, 0x62, 0x62, 0x60, 0x62, 0x62,
    0x60, 0x62, 0x60, 0x62, 0x62, 0x62, 0x62, 0x60,
    0x62, 0x60, 0x60, 0x60, 0x62, 0x62, 0x62, 0x60,
    0x60, 0x60, 0x62, 0x62, 0x62, 0x62, 0x62, 0x60
    };
```

3.5. Command Language

Table 3–9: Command table

Code Decimal	Code Hex	Command Name	No. Write Parameter	No. Read Parameter	Status Register
00	00	Dummy	0	0	x000 0000
01	01	Reset	0	0	x000 0000
02	02	Escape	0	0	x000 0000
03	03	Version	0	2	x000 0000
04	04	Test	0	0	x000 0000
05	05	Test	0	0	x000 0000
06	06	DRAM Mode	3	0	x000 0000
07	07	Acquisition Mode	5	2	x000 0000
08	08	Display Mode	3	0	x000 0000
09	09	Display TTX Pointer	2	0	x000 0000
10	0a	Display Pointer	3	0	x000 0000
11	0b	Display Clear	2	0	x000 0000
12	0c	Page Request	8	3	x0x0 0000
13	0d	Display Time Pointer	2	0	x000 0000
14	0e	Read DRAM Size	0	2	x000 0000
15	0f	Read VPS	0	15	x0x0 0000
16	10	Read Quality	0	4	x000 0000
17	11	Read Display Mode	0	3	x000 0000
18	12	Read Reset Source	0	1	x000 0000
19	13	Read Rolling Header	0	24	x000 0000
20	14	Read Page Info	2	4	x000 0000
21	15	Read Page Row	5	40	x0x0 0000
22	16	Change Page Info	3	0	x000 0000
23	17	Search MPET	0	1 + (n*4)	x0x0 0000
24	18	Read Display Page	0	4	x000 0000
25	19	Page Memory	2	0	x000 0000
26	1a	Display Page Request	5	0	x000 0000
27	1b	Page Table Reset	0	0	x000 0000
28	1c	Search Next Page	3	6	x0x0 0000
29	1d	Read Page Cycle	0	9	x000 0000
30	1e	Read TOP Code	2	2	x000 0000
31	1f	Read Rolling Time	0	8	x000 0000
32	20	Copy Page Row	8	0	x0x0 0000
33	21	Copy Data	7	0	x000 0000
34	22	Search Next TOP Code	3	4	x0x0 0000
35	23	Read Ghost Row	6	40	x0x0 0000
36	24	Read 8/30 Row	1	40	x0x0 0000
37	25	Read Priority	0	5	x000 0000
38	26	Page Priority	2	0	x000 0000
39	27	Search AIT	0	1 + (n*4)	x0x0 0000
40	28	Read TOP Status	0	2	x000 0000
41	29	Search AIT Title	2	17	x0x0 0000
42	2a	Reset Ghost Row Status	0	0	x000 0000
43	2b	Search MPT	0	1 + (n*4)	x0x0 0000
44	2c	Copy AIT Title	5	17	x0x0 0000
45	2d	Search Direct Choice	1	1 + (n*2)	x0x0 0000
46	2e	Read Hamming	1	1	x000 0000

TPU 3041

ADVANCE INFORMATION

Command table, continued

Code Decimal	Code Hex	Command Name	No. Write Parameter	No. Read Parameter	Status Register
47	2f	Read Hamming 2	3	3	x000 0000
48	30	Display Write Column	3+length	0	x000 0000
49	31	Display Fill	4	0	x000 0000
50	32	Read BTTL	0	9	x0x0 0000
51	33	Read Next Page	2	2	x000 0000
52	34	Change BTT magazine	1	0	x000 0000
53	35	Read WSS	0	15	x0x0 0000
54	36	Read CAPTION 1	0	7	x0x0 0000
55	37	Read CAPTION 2	0	7	x0x0 0000
56	38	OSD Font Pointer	5	0	x000 0000
57	39	Display Read Column	3	length	x000 0000
58	3a	User Character Set	8	0	x000 0000
59	3b	User ESC Character Set	8	0	x000 0000
60	3c	Full Row Attribute	3	0	x000 0000

12

ITT Semiconductors

Note:

If not otherwise designated all parameters in the following table are specified as single bytes. As write parameter magazine numbers 8 and 0 have the same meaning, a read parameter the magazine number is a true 4-bit number (e.g. magazine 8 = 00001000). For write parameters the values in parentheses indicate default values after reset (in hex notation). For compatibility reasons every undefined bit in a write parameter should be set to '0'. Undefined bits in a read parameter should be treated as "don't care".

Table 3–10: Command codes

Code	Function	Write Parameter		Read Parameter	Notes
		Operational & Test Commands			
00	Dummy				no action
01	Reset				software reset of 65C02
02	Escape				escape to other codes
03	Version			CPU pointer high CPU pointer low	show version in OSD layer CPU pointer to text in ROM
04	Test				reserved for testing
05	Test				reserved for testing

06	DRAM Mode	dram mode
		flash inc
		control enable
			(06)
			(05)
			(FF)
					dram mode = I/O page register 028EH
					flash freq = flash inc / (256 * 0.00324)
					control enable:
					    bit0 = C4 erase page
					    bit1 = C5 news flash
					    bit2 = C6 subtitle
					    bit3 = C7 suppress header
					    bit4 = C8 update indicator
					    bit5 = C9 interrupted sequence
					    bit6 = C10 inhibit display
					    bit7 = $\overline{C11}$ magazine parallel

07	Acquisition Mode	acquisition mode
		init subcode high
		init subcode low
		gain max
		filter max
			(00)
			(FF)
			(FF)
			(1F)
			(1F)
				gain
				filter
					acquisition mode:
					    bit0 = no slicer adaption
					    bit1 = no bit error in framing code
					    bit2 = limit slicer adaption
					init subcode:
					    automatic subcode request after
					    page table reset
					gain max: only used if bit2 = 1
					filter max: only used if bit2 = 1

		Memory Management Commands			

14	Read DRAM Size			dram size high
				dram size low
					dram size:
					    0080H = 256Kbit (slow mode)
					    0200H = 1Mbit (fast mode)
					    0800H = 4Mbit (fast mode)
					    2000H = 16Mbit (fast mode)

25	Page Memory	dram bank dram high	(00) (40)		start of page memory execute page table reset
27	Page Table Reset				reset page table reset ghost row status reset data service status reset cycle count reset memory count reset ghost count reset priorities clear rolling header clear VPS data clear WSS data

42	Reset Ghost Row Status				ghost row status:
					    bit0 = row 24 in cycle
					    bit1 = row 25 in cycle
					    bit2 = row 26 in cycle
					    bit3 = row 27 in cycle
					    bit4 = row 28 in cycle
					    bit5 = row 29 in cycle
					    bit6 = row 30 in cycle
					    bit7 = row 31 in cycle

Command codes, continued

Code	Function	Write Parameter	Read Parameter	Notes

29	Read Page Cycle		ghost row status
			2 byte cycle count
			2 byte memory count
			2 byte ghost count
			data service status
			memory status
				= number of pages in cycle
				= number of pages in memory
				= number of ghost blocks in memory
				data service status:
				    bit0 = 8/30 format 1 updated
				    bit1 = 8/30 format 2 updated
				    bit2 = VPS updated
				    bit3 = WSS updated
				    bit4 = CAPTION 1st field updated
				    bit5 = CAPTION 2nd field updated
				memory status:
				    bit0 = memory full

38	Page Priority	enable (00)
		border (FF)
				enable:
				    bit0 = enable priority manager
				border:
				    min/max border for page priorities

37	Read Priority		highest priority lowest priority border priority magazine number page number	= max priority in page memory = min priority in page memory = min/max border for page priorities = page with lowest priority
			Page Related Commands	

12	Page Request	magazine number
		page number
		page subcode high
		page subcode low
		priority
		quantity
		start magazine number
		start page number
			number of open requests
			removed magazine number
			removed page number
				remove pages from memory beginning at start page if page priority is disabled, ignores start page if page priority is enabled
				magazine number:
				    bit0–3 = magazine number
				    bit4 = not used
				    bit5 = hex request
				    bit6 = backward request
				    bit7 = forced request = ignore
				            cycle flag

20	Read Page Info	magazine number page number	page pointer high page pointer low subpage count ghost row count	= pointer from page table = number of subpages in chain = number of ghost rows in chain

22	Change Page Info	magazine number
		page number
		page table flags
				page table flags:
				    bit0 = protection
				    bit1 = update
				    bit2 = not used
				    bit3 = not used
				    bit4 = not used
				    bit5 = subpage
				    bit6 = memory
				    bit7 = cycle

28	Search Next Page	magazine number
		page number
		search code
			magazine number
			page number
			page pointer high
			page pointer low
			subpage count
			ghost row count
				search in page table for cycle flag
				magazine number:
				    bit0–3 = magazine number
				    bit4 = take search code
				    bit5 = hex search
				    bit6 = backward search
				    bit7 = include start page
				search code:
				    bit0 = search protection flag
				    bit1 = search update flag
				    bit2–4 = not used
				    bit5 = search subpage flag
				    bit6 = search memory flag
				    bit7 = search cycle flag

51	Read Next Page	magazine number
		page number
			magazine number
			page number
				calculate next page number
				magazine number:
				    bit0–3 = magazine number
				    bit4 = not used
				    bit5 = hex calculation
				    bit6 = backward calculation
				    bit7 = not used

Command codes, continued

Code	Function	Write Parameter	Read Parameter	Notes
21	Read Page Row	magazine number page number subpage number high subpage number low row number	40 byte row data	row 0 – 24
32	Copy Page Row	magazine number page number subpage number high subpage number low row number destination dram bank destination dram high destination dram low		copy 40byte text row from page memory into DRAM magazine number: bit0–3 = magazine number bit4–6 = not used bit7 = ignore language
35	Read Ghost Row	magazine number page number subpage number high subpage number low row number designation code	40 byte row data	row 25 – 28
TOP Commands				
40	Read TOP Status		TOP status 1 TOP status 2	TOP status 1: bit0 = not used bit1 = MPT link in PLT bit2 = MPET link in PLT bit3 = AIT link in PLT bit4 = BTT in memory bit5 = MPT in memory bit6 = MPET in memory bit7 = AIT in memory TOP status 2: bit0–5 = not used bit6 = all MPET in memory bit7 = all AIT in memory
30	Read TOP Code	magazine number page number	BTT code MPT code	code: bit0–3 = data bit6 = hamming error
50	Read BTTL		BTTL error 8 byte BTTL data	BTTL error: bit6 = hamming error in BTTL BTTL data: bit0–3 = data bit6 = hamming error
52	Change BTT magazine	magazine number (01)		all TOP commands are then referring to this magazine
43	Search MPT		number of MPTs magazine number page number subpage number high subpage number low ...	search in PLT
23	Search MPET		number of MPETs magazine number page number subpage number high subpage number low ...	search in PLT
39	Search AIT		number of AITs magazine number page number subpage number high subpage number low ...	search in PLT
41	Search AIT Title	magazine number page number	5 byte data 12 byte title	search in AIT magazine number: bit0–3 = magazine number (0#8) bit4–6 = not used bit7 = ignore title language data: bit0–3 = data bit6 = hamming error

TPU 3041

Command codes, continued

Code	Function	Write Parameter	Read Parameter	Notes
44	Copy AIT Title	magazine number page number destination dram bank destination dram high destination dram low	5 byte data 12 byte title	search in AIT and copy title into dram magazine number: bit0–3 = magazine number (0#8) bit4–6 = not used bit7 = ignore title language data: bit0–3 = data bit6 = hamming error
34	Search Next TOP Code	magazine number page number code condition	magazine number page number code code flag	search in BTT magazine number: bit0–3 = magazine number bit4–5 = not used bit6 = backward search bit7 = include start page code condition: low nibble = BTT code high nibble = search condition 0 = BTT code in low nibble 1 = BTT code # 0 2 = block page 3 = group page 4 = normal page 5 = subtitle page 6 = TV page 7 = block/TV page 8 = group/block/TV page 9 = subpage a = block/TV subpage b = group/block/TV subpage c = title page d = future page e = future page f = future page code: bit0–3 = BTT code bit6 = hamming error code flag: bit0 = subtitle page found bit1 = TV page found bit2 = block page found bit3 = group page found bit4 = normal page found bit5 = future page found bit6 = title page found bit7 = subpage found
45	Search Direct Choice	direct choice code	number of AIT entries magazine number page number ...	search in AIT

Command codes, continued

Code	Function	Write Parameter	Read Parameter	Notes
			Miscellaneous Data Commands	
36	Read 8/30 Row	designation code	40 byte row data	only format 1 and 2 are supported 1st byte of row data is already hamming decoded
15	Read VPS		framing code counter 13 byte VPS data	= 51H = incremented every VPS reception = biphase decoded VPS bytes 3–15
53	Read WSS		framing code counter 13 byte WSS data	= 78H = incremented every WSS reception = 102 WSS elements from group 1 on
54	Read CAPTION 1		counter 6 byte CAPTION data	= incremented every reception in field 1 = 3x oversampling
55	Read CAPTION 2		counter 6 byte CAPTION data	= incremented every reception in field 2 = 3x oversampling
19	Read Rolling Header		24 byte rolling header	every row 0 in cycle
31	Read Rolling Time		8 byte rolling time	using time pointer
16	Read Quality		text lines hamming errors parity errors soft errors	updated every VBI
18	Read Reset Source		reset source	reset source: bit0 = clock supervision bit1 = voltage supervision bit2 = watchdog all bits in reset source are reset after read
46	Read Hamming	hamming (8,4) byte	data	hamming byte: bit0–3 = data bit6 = hamming error
47	Read Hamming 2	hamming (24,18) 1st byte hamming (24,18) 2nd byte hamming (24,18) 3rd byte	address mode data	address: bit0–5 = address bit7 = hamming error mode: bit0–4 = mode data: bit0–6 = data
33	Copy Data	source dram bank source dram high source dram low length destination dram bank destination dram high destination dram low		copy data from DRAM to DRAM
			Display Commands	
17	Read Display Mode		display mode character set font	display mode: bit0 = forced boxing bit1 = reveal bit2 = box bit3 = time hold bit4 = page hold bit5 = row 24 hold bit6 = row 25 hold bit7 = row 26 hold
08	Display Mode	display mode (00) character set (06) font (00)		display mode: see above character set: 6,38,40,55,70,128 font: not used
09	Display TTX Pointer	dram high (20) dram low (00)		page memory is copied to TTX pointer
10	Display Pointer	dram high (20) dram low (00) scroll counter (00)		display starts at pointer using scroll counter as line offset
11	Display Clear	dram high dram low		clear display bank beginning at pointer (26 rows * 86 bytes)
13	Display Time Pointer	dram high (20) dram low (20)		8byte time string from packet x/00 is copied to time pointer

Command codes, continued

Code	Function	Write Parameter	Read Parameter	Notes

26	Display Page Request	magazine number
		page number
		subpage number high
		subpage number low
		display delay (1E)
				magazine number:
				    bit0–3 = magazine number
				    bit4 = change display delay
				    bit5 = display clear (on update)
				    bit6–7 = not used
				subpage number:
				    F0xx for rolling subpages
				display delay:
				    delay after row 0 reception in steps
				    of 3.24ms (255 = no update)
				    only used if bit4 = 1

24	Read Display Page		magazine number page number subpage number high subpage number low	current page in display
48	Display Write Column	dram high dram low length byte list ...		write to dram with increment of 86 byte = number of bytes in list
49	Display Fill	dram high dram low length character		repeated write of 1 character to dram = number of repeated writes

56	OSD Font Pointer	font mode (00)
		font pointer high
		font pointer low
		extension font pointer high
		extension font pointer low
				font mode:
				    bit0 = 0 = reset OSD font 2 pointer
				    bit0 = 1 = load OSD font 2 pointer
				        with following parameters

57	Display Read Column	dram high dram low length	byte list ...	read from dram with increment of 86 byte = number of bytes to read
58	User Character Set	language 000 (00) language 001 (01) language 010 (02) language 011 (03) language 100 (04) language 101 (05) language 110 (06) language 111 (07)		If character set 128 is selected via command 08 "Display Mode", these 8 languages will be selected by option bits C14,C13,C12 when ESC code is inactive.
59	User ESC Character Set	esc language 000 (00) esc language 001 (00) esc language 010 (00) esc language 011 (00) esc language 100 (00) esc language 101 (00) esc language 110 (00) esc language 111 (00)		If character set 128 is selected via command 08 "Display Mode", these 8 languages will be selected by option bits C14,C13,C12 when ESC code is active.
60	Full Row Attribute	full row attribute number of rows start row		set full row attribute of specified rows without changing level 2 bit

4. Data Sheet History

1. Advance information: "TPU 3041
Cyrillic/Greek Teletext Processor", June 12, 1995,
6251-415-2AI. Second release of advance information.

2. Advance information: "TPU 3041
Cyrillic/Greek Teletext Processor", Sept. 27, 1995,
6251-415-3AI. Third release of advance information.

End of Data Sheet

Back to Summary ◄

Back to Data Sheets ICs ◄

ITT

Semiconductors