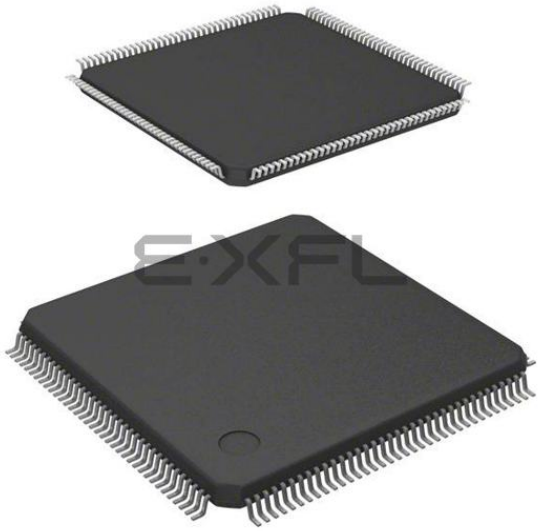


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	ST10
Core Size	16-Bit
Speed	40MHz
Connectivity	ASC, CANbus, EBI/EMI, I²C, SSC, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	111
Program Memory Size	832KB (832K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	68K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 24x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st10f276-4t3

4.5 Write operation examples

In this section, examples for each kind of Flash write operation are presented.

Word program

Example: 32-bit Word Program of data 0xAAAAAAAA at address 0x0C5554 in XFLASH Module.

```
FCR0H|= 0x2000; /*Set WPG in FCR0H*/
FARL = 0x5554; /*Load Add in FARL*/
FARH = 0x000C; /*Load Add in FARH*/
FDR0L = 0xAAAA; /*Load Data in FDR0L*/
FDR0H = 0xAAAA; /*Load Data in FDR0H*/
FCR0H|= 0x8000; /*Operation start*/
```

Double word program

Example: Double Word Program (64-bit) of data 0x55AA55AA at address 0x095558 and data 0xAA55AA55 at address 0x09555C in IFLASH Module.

```
FCR0H|= 0x1080; /*Set DWPG, SMOD*/
FARL  = 0x5558; /*Load Add in FARL*/
FARH  = 0x0009; /*Load Add in FARH*/
FDR0L = 0x55AA; /*Load Data in FDR0L*/
FDR0H = 0x55AA; /*Load Data in FDR0H*/
FDR1L = 0xAA55; /*Load Data in FDR1L*/
FDR1H = 0xAA55; /*Load Data in FDR1H*/
FCR0H|= 0x8000; /*Operation start*/
```

Double Word Program is always performed on the Double Word aligned on a even Word: bit ADD2 of FARL is ignored.

Sector erase

Example: Sector Erase of sectors B3F1 and B3F0 of Bank 3 in XFLASH Module.

```
FCR0H|= 0x0800; /*Set SER in FCR0H*/
FCR1H|= 0x0003; /*Set B3F1, B3F0*/
FCR0H|= 0x8000; /*Operation start*/
```

Suspend and resume

Word Program, Double Word Program, and Sector Erase operations can be suspended in the following way:

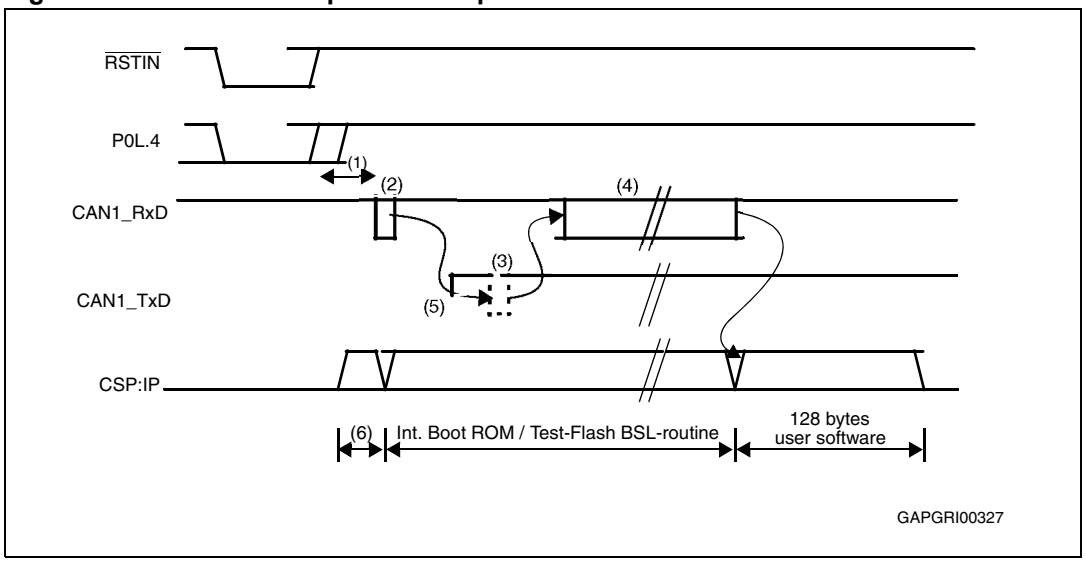
```
FCR0H|= 0x4000; /*Set SUSP in FCR0H*/
```

Then the operation can be resumed in the following way:

```
FCR0H|= 0x0800; /*Set SER in FCR0H*/
FCR0H|= 0x8000; /*Operation resume*/
```

Before resuming a suspended Erase, FCR1H/FCR1L must be read to check if the Erase is already completed (FCR1H = FCR1L = 0x0000 if Erase is complete). Original setup of Select Operation bits in FCR0H/L must be restored before the operation resume, otherwise the operation is aborted and bit RESER of FER is set.

Figure 11. CAN bootstrap loader sequence



1. BSL initialization time, > 1ms @ f_{CPU} = 40 MHz
2. Zero frame (CAN message: standard ID = 0, DLC = 0), sent by host
3. CAN message (standard ID = E6h, DLC = 3, Data0 = D5h, Data1-Data2 = IDCHIP_low-high), sent by ST10F276E on request
4. 128 bytes of code / data, sent by host
5. Caution: CAN1_TxD is only driven a certain time after reception of the zero byte (1.3ms @ f_{CPU} = 40 MHz).
6. Internal Boot ROM / Test-Flash

The Bootstrap Loader can load

- the complete application software into ROM-less systems,
- temporary software into complete systems for testing or calibration,
- a programming routine for Flash devices.

The BSL mechanism may be used for standard system start-up as well as for only special occasions like system maintenance (firmware update) or end-of-line programming or testing.

5.4.2 Entering the CAN bootstrap loader

The ST10F276E enters BSL mode if pin P0L.4 is sampled low at the end of a hardware reset. In this case, the built-in bootstrap loader is activated independently of the selected bus mode. The bootstrap loader code is stored in a special Test-Flash; no part of the standard mask ROM or Flash memory area is required for this.

After entering BSL mode and the respective initialization, the ST10F276E scans the CAN1_TxD line to receive the following initialization frame:

- Standard identifier = 0h
- DLC = 0h

As all the bits to be transmitted are dominant bits, a succession of 5 dominant bits and 1 stuff bit on the CAN network is used. From the duration of this frame, it calculates the corresponding baud rate factor with respect to the current CPU clock, initializes the CAN1 interface accordingly, switches pin CAN1_TxD to output and enables the CAN1 interface to take part in the network communication. Using this baud rate, a Message Object is

5.4.3 ST10 configuration in CAN BSL

When the ST10F276E enters BSL mode via CAN, the configuration shown in *Table 31* is automatically set (values that deviate from the normal reset values are marked in **bold**).

Table 31. ST10 configuration in CAN BSL

Function or register	Access	Notes
Watchdog timer	Disabled	
Register SYSCON	**0404_H** (1)	XPEN bit set
Context pointer CP	$FA00_H$	
Register STKUN	$FA00_H$	
Stack pointer SP	$FA40_H$	
Register STKOV	$FC00_H$	
Register BUSCON0	acc. to startup config.(2)	
CAN1 Status/Control register	0000_H	Initialized only if Bootstrap via CAN
CAN1 Bit timing register	acc. to '0' frame	Initialized only if Bootstrap via CAN
XPERCON	$042D_H$	XRAM1-2, XFlash, CAN1 and XMISC enabled
P4.6 / CAN1_TxD	**'1'**	Initialized only if Bootstrap via CAN
DP4.6	**'1'**	Initialized only if Bootstrap via CAN

1. In Bootstrap modes (standard or alternate) ROMEN, bit 10 of SYSCON, is always set regardless of $\overline{EA}$ pin level. BYTDIS, bit 9 of SYSCON, is set according to data bus width selection via Port0 configuration.
2. BUSCON0 is initialized with 0000h, external bus disabled, if pin $\overline{EA}$ is high during reset. If pin $\overline{EA}$ is low during reset, BUSACT0, bit 10, and ALECTL0, bit 9, are set enabling the external bus with lengthened ALE signal. BTYP field, bit 7 and 6, is set according to Port0 configuration.

Other than after a normal reset, the watchdog timer is disabled, so the bootstrap loading sequence is not time limited. Pin CAN1_TxD1 is configured as output, so the ST10F276E can return the identification frame. Even if the internal IFLASH is enabled, a code cannot be executed from it.

5.4.4 Loading the start-up code via CAN

After sending the acknowledge byte, the BSL enters a loop to receive 128 bytes via CAN1.

Hint: The number of bytes loaded when booting via the CAN interface has been extended to 128 bytes to allow the reconfiguration of the CAN Bit Timing Register with the best timings (synchronization window, ...). This can be achieved by the following sequence of instructions:

```
ReconfigureBaud rate:
  MOV  R1,#041h
  MOV  DPP3:0EF00h,R1; Put CAN in Init, enable Configuration Change
  MOV  R1,#01600h
  MOV  DPP3:0EF06h,R1; 1MBaud at Fcpu = 20 MHz
```

These 128 bytes are stored sequentially into locations 00'$FA40_H$ through 00'$FABF_H$ of the IRAM, allowing up to 64 instructions to be placed into the RAM area. To execute the loaded code the BSL then jumps to location 00'$FA40_H$, that is, the first loaded instruction. The

19 System reset

System reset initializes the MCU in a predefined state. There are six ways to activate a reset state. The system start-up configuration is different for each case as shown in *Table 60*.

Table 60. Reset event definition

Reset source	Flag	RPD status	Conditions
Power-on reset	PONR	Low	Power-on
Asynchronous Hardware reset	LHWR	Low	$t_{\overline{RSTIN}}$ >(1)
Synchronous Long Hardware reset		High	$t_{\overline{RSTIN}}$ > (1032 + 12) TCL + max(4 TCL, 500ns)(2)
Synchronous Short Hardware reset	SHWR	High	$t_{\overline{RSTIN}}$ > max(4 TCL, 500ns) $t_{\overline{RSTIN}}$ ≤ (1032 + 12) TCL + max(4 TCL, 500ns)(2)
Watchdog Timer reset	WDTR	(3)	WDT overflow
Software reset	SWR	(3)	SRST instruction execution

1. $\overline{RSTIN}$ pulse should be longer than 500ns (Filter) and than settling time for configuration of Port0.
2. See next *Section 19.1* for more details on minimum reset pulse duration.
3. The RPD status has no influence unless Bidirectional Reset is activated (bit BDRSTEN in SYSCON): RPD low inhibits the Bidirectional reset on SW and WDT reset events, that is $\overline{RSTIN}$ is not activated (refer to sections *19.4*, *19.5* and *19.6*).

19.1 Input filter

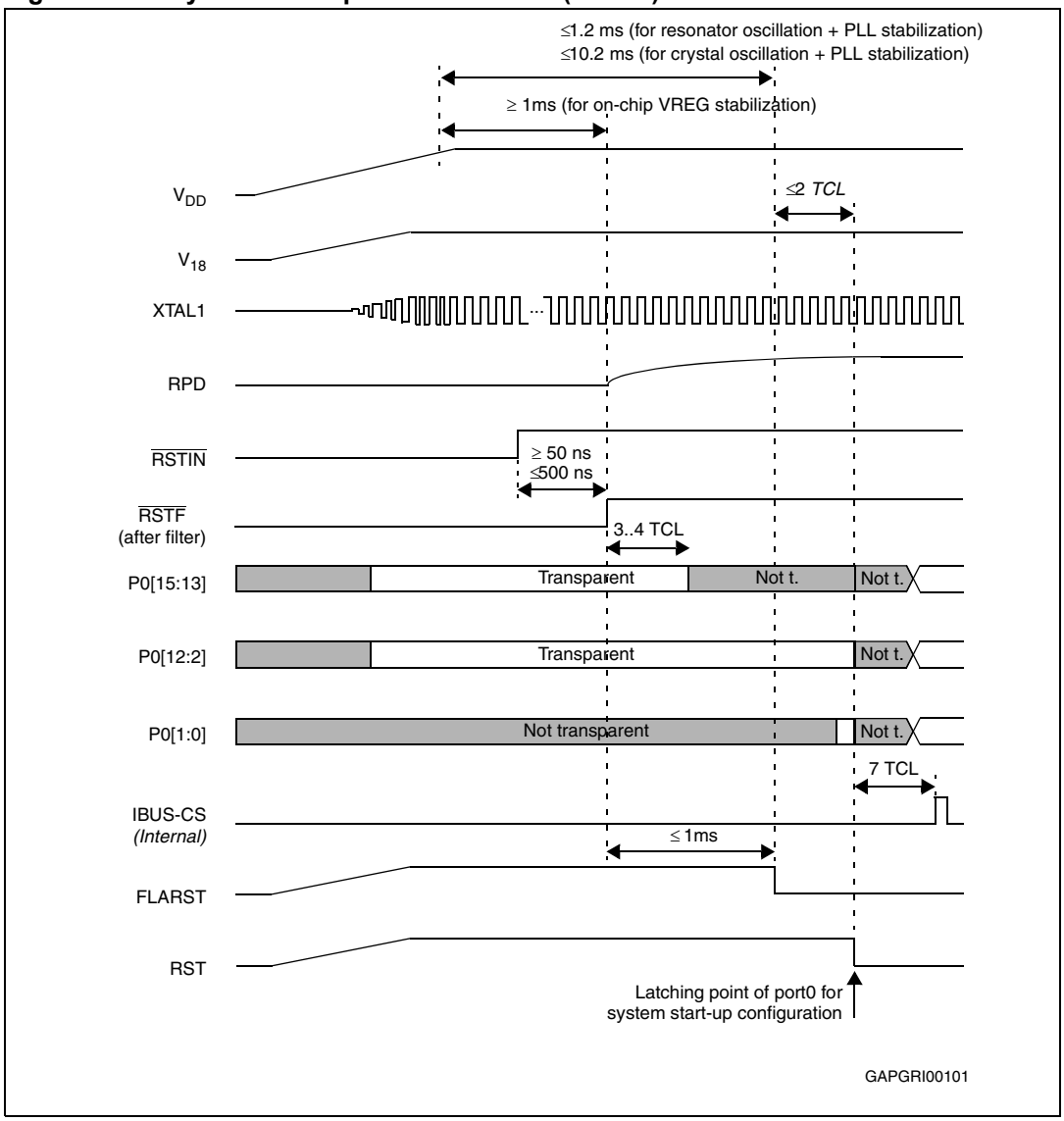
On $\overline{RSTIN}$ input pin an on-chip RC filter is implemented. It is sized to filter all the spikes shorter than 50ns. On the other side, a valid pulse shall be longer than 500ns to grant that ST10 recognizes a reset command. In between 50ns and 500ns a pulse can either be filtered or recognized as valid, depending on the operating conditions and process variations.

For this reason all minimum durations mentioned in this Chapter for the different kind of reset events shall be carefully evaluated taking into account of the above requirements.

In particular, for Short Hardware Reset, where only 4 TCL is specified as minimum input reset pulse duration, the operating frequency is a key factor. Examples:

- For a CPU clock of 64 MHz, 4 TCL is 31.25ns, so it would be filtered. In this case the minimum becomes the one imposed by the filter (that is 500ns).
- For a CPU clock of 4 MHz, 4 TCL is 500ns. In this case the minimum from the formula is coherent with the limit imposed by the filter.

Figure 24. Asynchronous power-on RESET ($\overline{EA}$ = 1)



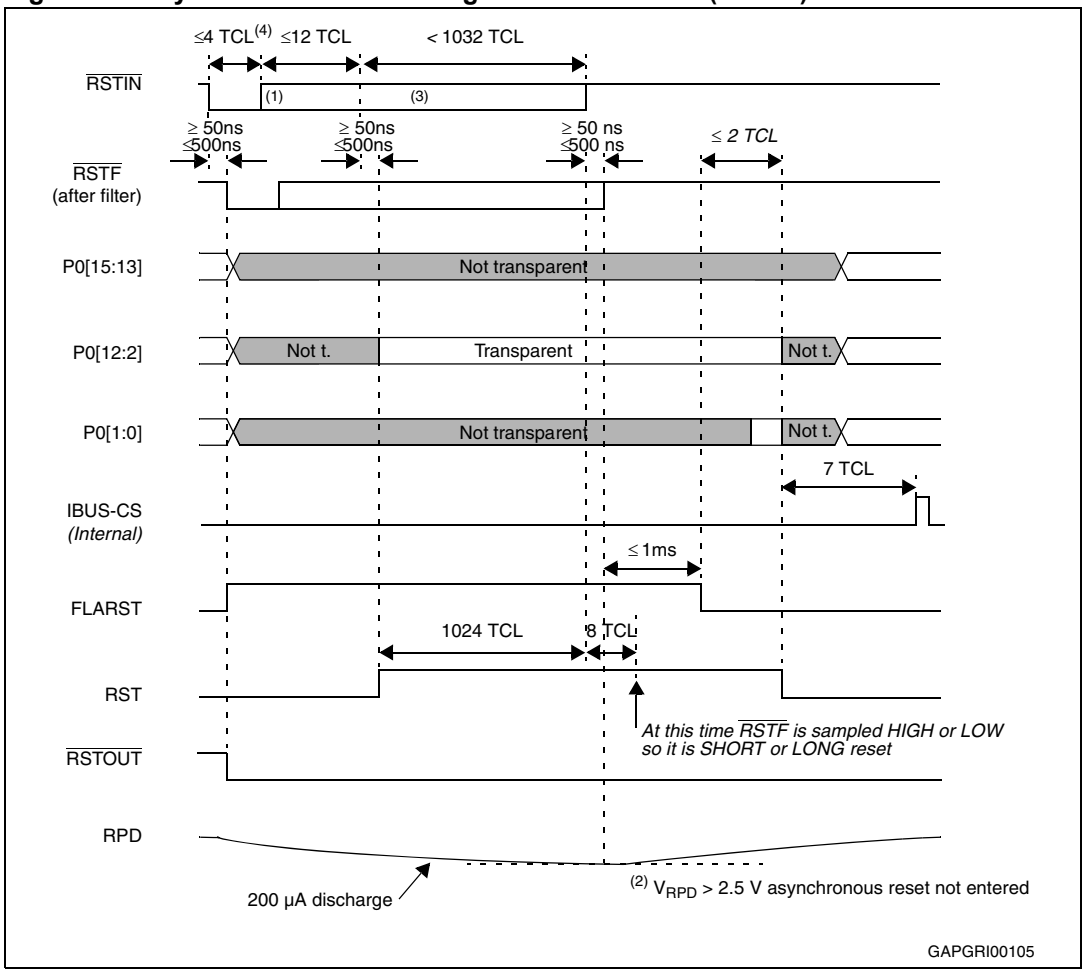
Synchronous reset and RPD pin

Whenever the $\overline{\text{RSTIN}}$ pin is pulled low (by external hardware or as a consequence of a Bidirectional reset), the RPD internal weak pull-down is activated. The external capacitance (if any) on RPD pin is slowly discharged through the internal weak pull-down. If the voltage level on RPD pin reaches the input low threshold (around 2.5V), the reset event becomes immediately asynchronous. In case of hardware reset (short or long) the situation goes immediately to the one illustrated in *Figure 26*. There is no effect if RPD comes again above the input threshold: the asynchronous reset is completed coherently. To grant the normal completion of a synchronous reset, the value of the capacitance shall be big enough to maintain the voltage on RPD pin sufficient high along the duration of the internal reset sequence.

For a Software or Watchdog reset event, an active synchronous reset is completed regardless of the RPD status.

It is important to highlight that the signal that makes RPD status transparent under reset is the internal $\overline{\text{RSTF}}$ (after the noise filter).

Figure 28. Synchronous short / long hardware RESET ($\overline{\text{EA}}$ = 1)



1. RSTIN assertion can be released there. Refer also to *Section 19.1* for details on minimum pulse duration.
2. If during the reset condition (RSTIN low), RPD voltage drops below the threshold voltage (about 2.5V for

Note: 1 The system configuration is selected during reset. SYSCON reset value is 0000 0xx0 x000 0000b.

2 Reset Value depends on different triggered reset event.

3 The XPnIC Interrupt Control Registers control interrupt requests from integrated X-Bus peripherals. Some software controlled interrupt requests may be generated by setting the XPnIR bits (of XPnIC register) of the unused X-Peripheral nodes.

22.5 Special function registers ordered by address

Table 68 lists by order of their physical addresses all SFRs which are implemented in the ST10F276E.

Bit-addressable SFRs are marked with the letter "**b**" in column "Name".

SFRs within the **Extended SFR-Space** (ESFRs) are marked with the letter "**E**" in column "Physical Address".

Table 68. Special function registers ordered by address

Name	Physical address	8-bit address	Description	Reset value
QX0	F000h**E**	00h	MAC unit offset register X0	0000h
QX1	F002h**E**	01h	MAC unit offset register X1	0000h
QR0	F004h**E**	02h	MAC unit offset register R0	0000h
QR1	F006h**E**	03h	MAC unit offset register R1	0000h
XADRS3	F01Ch**E**	0Eh	XPER address select register 3	800Bh
XPERCON	F024h**E**	12h	XPER configuration register	- - 05h
PT0	F030h**E**	18h	PWM module up/down counter 0	0000h
PT1	F032h**E**	19h	PWM module up/down counter 1	0000h
PT2	F034h**E**	1Ah	PWM module up/down counter 2	0000h
PT3	F036h**E**	1Bh	PWM module up/down counter 3	0000h
PP0	F038h**E**	1Ch	PWM module period register 0	0000h
PP1	F03Ah**E**	1Dh	PWM module period register 1	0000h
PP2	F03Ch**E**	1Eh	PWM module period register 2	0000h
PP3	F03Eh**E**	1Fh	PWM module period register 3	0000h
T7	F050h**E**	28h	CAPCOM timer 7 register	0000h
T8	F052h**E**	29h	CAPCOM timer 8 register	0000h
T7REL	F054h**E**	2Ah	CAPCOM timer 7 reload register	0000h
T8REL	F056h**E**	2Bh	CAPCOM timer 8 reload register	0000h
IDPROG	F078h**E**	3Ch	Programming voltage identifier register	0040h
IDMEM	F07Ah**E**	3Dh	On-chip memory identifier register	30D0h
IDCHIP	F07Ch**E**	3Eh	Device identifier register (n is the device revision)	114nh

Table 68. Special function registers ordered by address (continued)

Name	Physical address	8-bit address	Description	Reset value
XP3IC **b**	F19Eh**E**	CFh	See *Section 8.1*	- - 00h
EXICON **b**	F1C0h**E**	E0h	External interrupt control register	0000h
ODP2 **b**	F1C2h**E**	E1h	Port2 open drain control register	0000h
PICON **b**	F1C4h**E**	E2h	Port input threshold control register	- - 00h
ODP3 **b**	F1C6h**E**	E3h	Port3 open drain control register	0000h
ODP4 **b**	F1CAh**E**	E5h	Port4 open drain control register	- - 00h
ODP6 **b**	F1CEh**E**	E7h	Port6 open drain control register	- - 00h
ODP7 **b**	F1D2h**E**	E9h	Port7 open drain control register	- - 00h
ODP8 **b**	F1D6h**E**	EBh	Port8 open drain control register	- - 00h
EXISEL **b**	F1DAh**E**	EDh	External interrupt source selection register	0000h
DPP0	FE00h	00h	CPU data page pointer 0 register (10-bit)	0000h
DPP1	FE02h	01h	CPU data page pointer 1 register (10-bit)	0001h
DPP2	FE04h	02h	CPU data page pointer 2 register (10-bit)	0002h
DPP3	FE06h	03h	CPU data page pointer 3 register (10-bit)	0003h
CSP	FE08h	04h	CPU code segment pointer register (read-only)	0000h
EMUCON	FE0Ah	05h	Emulation control register	- - XXh
MDH	FE0Ch	06h	CPU multiply divide register – High word	0000h
MDL	FE0Eh	07h	CPU multiply divide register – Low word	0000h
CP	FE10h	08h	CPU context pointer register	FC00h
SP	FE12h	09h	CPU system stack pointer register	FC00h
STKOV	FE14h	0Ah	CPU stack overflow pointer register	FA00h
STKUN	FE16h	0Bh	CPU stack underflow pointer register	FC00h
ADDRSEL1	FE18h	0Ch	Address select register 1	0000h
ADDRSEL2	FE1Ah	0Dh	Address select register 2	0000h
ADDRSEL3	FE1Ch	0Eh	Address select register 3	0000h
ADDRSEL4	FE1Eh	0Fh	Address select register 4	0000h
PW0	FE30h	18h	PWM module pulse width register 0	0000h
PW1	FE32h	19h	PWM module pulse width register 1	0000h
PW2	FE34h	1Ah	PWM module pulse width register 2	0000h
PW3	FE36h	1Bh	PWM module pulse width register 3	0000h
T2	FE40h	20h	GPT1 timer 2 register	0000h
T3	FE42h	21h	GPT1 timer 3 register	0000h
T4	FE44h	22h	GPT1 timer 4 register	0000h

Table 68. Special function registers ordered by address (continued)

Name	Physical address	8-bit address	Description	Reset value
SSCEIC **b**	FF76h	BBh	SSC error interrupt control register	- - 00h
CC0IC **b**	FF78h	BCh	CAPCOM register 0 interrupt control register	- - 00h
CC1IC **b**	FF7Ah	BDh	CAPCOM register 1 interrupt control register	- - 00h
CC2IC **b**	FF7Ch	BEh	CAPCOM register 2 interrupt control register	- - 00h
CC3IC **b**	FF7Eh	BFh	CAPCOM register 3 interrupt control register	- - 00h
CC4IC **b**	FF80h	C0h	CAPCOM register 4 interrupt control register	- - 00h
CC5IC **b**	FF82h	C1h	CAPCOM register 5 interrupt control register	- - 00h
CC6IC **b**	FF84h	C2h	CAPCOM register 6 interrupt control register	- - 00h
CC7IC **b**	FF86h	C3h	CAPCOM register 7 interrupt control register	- - 00h
CC8IC **b**	FF88h	C4h	CAPCOM register 8 interrupt control register	- - 00h
CC9IC **b**	FF8Ah	C5h	CAPCOM register 9 interrupt control register	- - 00h
CC10IC **b**	FF8Ch	C6h	CAPCOM register 10 interrupt control register	- - 00h
CC11IC **b**	FF8Eh	C7h	CAPCOM register 11 interrupt control register	- - 00h
CC12IC **b**	FF90h	C8h	CAPCOM register 12 interrupt control register	- - 00h
CC13IC **b**	FF92h	C9h	CAPCOM register 13 interrupt control register	- - 00h
CC14IC **b**	FF94h	CAh	CAPCOM register 14 interrupt control register	- - 00h
CC15IC **b**	FF96h	CBh	CAPCOM register 15 interrupt control register	- - 00h
ADCIC **b**	FF98h	CCh	A/D converter end of conversion interrupt control register	- - 00h
ADEIC **b**	FF9Ah	CDh	A/D converter overrun error interrupt control register	- - 00h
T0IC **b**	FF9Ch	CEh	CAPCOM timer 0 interrupt control register	- - 00h
T1IC **b**	FF9Eh	CFh	CAPCOM timer 1 interrupt control register	- - 00h
ADCON **b**	FFA0h	D0h	A/D converter control register	0000h
P5 **b**	FFA2h	D1h	Port 5 register (read-only)	XXXXh
P5DIDIS **b**	FFA4h	D2h	Port 5 digital disable register	0000h
TFR **b**	FFACh	D6h	Trap flag register	0000h
WDTCON **b**	FFAEh	D7h	Watchdog timer control register	00xxh
S0CON **b**	FFB0h	D8h	Serial channel 0 control register	0000h
SSCCON **b**	FFB2h	D9h	SSC control register	0000h
P2 **b**	FFC0h	E0h	Port 2 register	0000h
DP2 **b**	FFC2h	E1h	Port 2 direction control register	0000h
P3 **b**	FFC4h	E2h	Port 3 register	0000h
DP3 **b**	FFC6h	E3h	Port 3 direction control register	0000h

Table 69. X-Registers ordered by name (continued)

Name	Physical address	Description	Reset value
CAN1IF1M2	EF16h	CAN1: IF1 mask 2	FFFFh
CAN1IF1MC	EF1Ch	CAN1: IF1 message control	0000h
CAN1IF2A1	EF48h	CAN1: IF2 arbitration 1	0000h
CAN1IF2A2	EF4Ah	CAN1: IF2 arbitration 2	0000h
CAN1IF2CM	EF42h	CAN1: IF2 command mask	0000h
CAN1IF2CR	EF40h	CAN1: IF2 command request	0001h
CAN1IF2DA1	EF4Eh	CAN1: IF2 data A 1	0000h
CAN1IF2DA2	EF50h	CAN1: IF2 data A 2	0000h
CAN1IF2DB1	EF52h	CAN1: IF2 data B 1	0000h
CAN1IF2DB2	EF54h	CAN1: IF2 data B 2	0000h
CAN1IF2M1	EF44h	CAN1: IF2 mask 1	FFFFh
CAN1IF2M2	EF46h	CAN1: IF2 mask 2	FFFFh
CAN1IF2MC	EF4Ch	CAN1: IF2 message control	0000h
CAN1IP1	EFA0h	CAN1: interrupt pending 1	0000h
CAN1IP2	EFA2h	CAN1: interrupt pending 2	0000h
CAN1IR	EF08h	CAN1: interrupt register	0000h
CAN1MV1	EFB0h	CAN1: Message valid 1	0000h
CAN1MV2	EFB2h	CAN1: Message valid 2	0000h
CAN1ND1	EF90h	CAN1: New data 1	0000h
CAN1ND2	EF92h	CAN1: New data 2	0000h
CAN1SR	EF02h	CAN1: Status register	0000h
CAN1TR	EF0Ah	CAN1: Test register	00x0h
CAN1TR1	EF80h	CAN1: Transmission request 1	0000h
CAN1TR2	EF82h	CAN1: Transmission request 2	0000h
CAN2BRPER	EE0Ch	CAN2: BRP extension register	0000h
CAN2BTR	EE06h	CAN2: Bit timing register	2301h
CAN2CR	EE00h	CAN2: CAN control register	0001h
CAN2EC	EE04h	CAN2: Error counter	0000h
CAN2IF1A1	EE18h	CAN2: IF1 arbitration 1	0000h
CAN2IF1A2	EE1Ah	CAN2: IF1 arbitration 2	0000h
CAN2IF1CM	EE12h	CAN2: IF1 command mask	0000h
CAN2IF1CR	EE10h	CAN2: IF1 command request	0001h
CAN2IF1DA1	EE1Eh	CAN2: IF1 data A 1	0000h
CAN2IF1DA2	EE20h	CAN2: IF1 data A 2	0000h

Table 69. X-Registers ordered by name (continued)

Name	Physical address	Description	Reset value
XSSCRB	E808h	XSSC receive buffer	XXXXh
XSSCTB	E806h	XSSC transmit buffer	0000h

Note: 1 BTYP (bit 6 and 7) is set according to the configuration of the bit I6 and I7 of PORT0 latched at the end of the reset sequence.

2 BUSCON0 is initialized with 0000h, if $\overline{EA}$ pin is high during reset. If $\overline{EA}$ pin is low during reset, bit BUSACT0 and ALECTRL0 are set ('1') and bit field BTYP is loaded with the bus configuration selected via PORT0.

RP0H (F108h / 84h) ESFR Reset value: --XXh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-								CLKSEL			SALSEL		CSSEL		WRC
								R			R		R		R

Table 79. RPOH description(1)

Bit	Function
WRC(2)	Write configuration control 0: Pin $\overline{WR}$ acts as $\overline{WRL}$, pin $\overline{BHE}$ acts as $\overline{WRH}$ 1: Pins $\overline{WR}$ and $\overline{BHE}$ retain their normal function
CSSEL(2)	Chip select line selection (number of active $\overline{CS}$ outputs) 00: 3 $\overline{CS}$ lines: $\overline{CS}$2...$\overline{CS}$0 01: 2 $\overline{CS}$ lines: $\overline{CS}$1...$\overline{CS}$0 10: No $\overline{CS}$ line at all 11: 5 $\overline{CS}$ lines: $\overline{CS}$4...$\overline{CS}$0 (Default without pull-downs)
SALSEL(2)	Segment address line selection (number of active segment address outputs) 00: 4-bit segment address: A19...A16 01: No segment address lines at all 10: 8-bit segment address: A23...A16 11: 2-bit segment address: A17...A16 (Default without pull-downs)
CLKSEL(2)(3)	System clock selection 000: $f_{CPU} = 16 \times f_{OSC}$ 001: $f_{CPU} = 0.5 \times f_{OSC}$ 010: $f_{CPU} = 10 \times f_{OSC}$ 011: $f_{CPU} = f_{OSC}$ 100: $f_{CPU} = 5 \times f_{OSC}$ 101: $f_{CPU} = 8 \times f_{OSC}$ 110: $f_{CPU} = 3 \times f_{OSC}$ 111: $f_{CPU} = 4 \times f_{OSC}$

1. RP0H is a read-only register.
2. These bits are set according to Port 0 configuration during any reset sequence.
3. RP0H.7 to RP0H.5 bits are loaded only during a long hardware reset. As pull-up resistors are active on each Port P0H pins during reset, RP0H default value is "FFh".

EXICON (F1C0h / E0h) ESFR Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXI7ES		EXI6ES		EXI5ES		EXI4ES		EXI3ES		EXI2ES		EXI1ES		EXI0ES	
RW		RW		RW		RW		RW		RW		RW		RW	

Table 84. ESFR description (continued)

Bit	Function
XRAM2EN	XRAM2 enable bit 0: Accesses to the on-chip 64 Kbyte XRAM are disabled, external access performed. Address range 0F'0000h-0F'FFFFh is directed to external memory only if XFLASHEN is '0' also. 1: The on-chip 64 Kbyte XRAM is enabled and can be accessed.
XRTCEN	RTC enable 0: Accesses to the on-chip RTC module are disabled, external access performed. Address range 00'ED00h-00'EDFF is directed to external memory only if CAN1EN, CAN2EN, XASCEN, XSSCEN, XI2CEN, XPWMEN and XMISCEN are '0' also. 1: The on-chip RTC module is enabled and can be accessed.
XPWMEN	XPWM enable 0: Accesses to the on-chip XPWM module are disabled, external access performed. Address range 00'EC00h-00'ECFF is directed to external memory only if CAN1EN, CAN2EN, XASCEN, XSSCEN, XI2CEN, XRTCEN and XMISCEN are '0' also. 1: The on-chip XPWM module is enabled and can be accessed.
XFLASHEN	XFlash enable bit 0: Accesses to the on-chip XFlash and Flash registers are disabled, external access performed. Address range 09'0000h-0E'FFFFh is directed to external memory only if XRAM2EN is '0' also. 1: The on-chip XFlash is enabled and can be accessed.
XASCEN	XASC enable bit 0: Accesses to the on-chip XASC are disabled, external access performed. Address range 00'E900h-00'E9FFh is directed to external memory only if CAN1EN, CAN2EN, XRTCEN, XASCEN, XI2CEN, XPWMEN and XMISCEN are '0' also. 1: The on-chip XASC is enabled and can be accessed.
XSSCEN	XSSC enable bit 0: Accesses to the on-chip XSSC are disabled, external access performed. Address range 00'E800h-00'E8FFh is directed to external memory only if CAN1EN, CAN2EN, XRTCEN, XASCEN, XI2CEN, XPWMEN and XMISCEN are '0' also. 1: The on-chip XSSC is enabled and can be accessed.
XI2CEN	I^2C enable bit 0: Accesses to the on-chip I^2C are disabled, external access performed. Address range 00'EA00h-00'EAFFh is directed to external memory only if CAN1EN, CAN2EN, XRTCEN, XASCEN, XSSCEN, XPWMEN and XMISCEN are '0' also. 1: The on-chip I^2C is enabled and can be accessed.
XMISCEN	XBUS additional features enable bit 0: Accesses to the Additional Miscellaneous Features is disabled. Address range 00'EB00h-00'EBFFh is directed to external memory only if CAN1EN, CAN2EN, XRTCEN, XASCEN, XSSCEN, XPWMEN and XI2CEN are '0' also. 1: The Additional Features are enabled and can be accessed.

When CAN1, CAN2, RTC, XASC, XSSC, I^2C, XPWM and the XBUS Additional Features are all disabled via XPERCON setting, then any access in the address range 00'E800h - 00'EFFFh is directed to external memory interface, using the BUSCONx register corresponding to the address matching ADDRSELx register. All pins used for X-Peripherals can be used as General Purpose I/O whenever the related module is not enabled.

The default XPER selection after Reset is such that CAN1 is enabled, CAN2 is disabled, XRAM1 (2 Kbyte XRAM) is enabled and XRAM2 (64 Kbyte XRAM) is disabled; all the other X-Peripherals are disabled after Reset.

Register XPERCON cannot be changed after the global enabling of X-Peripherals, that is, after setting of bit XPEN in SYSCON register.

In Emulation mode, all the X-Peripherals are enabled (XPERCON bits are all set). The bondout chip determines whether or not to redirect an access to external memory or to XBUS.

Reserved bits of XPERCON register are always written to '0'.

Table 85 below summarizes the Segment 8 mapping that depends upon the $\overline{EA}$ pin status during reset as well as the SYSCON (bit XPEN) and XPERCON (bits XRAM2EN and XFLASHEN) registers user programmed values.

Table 85. Segment 8 address range mapping

$\overline{EA}$	XPEN	XRAM2EN	XFLASHEN	Segment 8
0	0	x	x	External memory
0	1	0	0	External memory
0	1	1	x	Reserved
0	1	x	1	Reserved
1	x	x	x	IFlash (B1F1)

Note: The symbol "x" in the table above stands for "do not care".

22.12 XPERCON and XPEREMU registers

As already mentioned, the XPERCON register must be programmed to enable the single XBUS modules separately. The XPERCON is a read/write ESFR register; the XPEREMU register is a write-only register mapped on XBUS memory space (address EB7Eh).

Once the XPEN bit of SYSCON register is set and at least one of the X-peripherals (except memories) is activated, the register XPEREMU must be written with the same content of XPERCON: This is mandatory in order to allow a correct emulation of the new set of features introduced on XBUS for the new ST10 generation. The following instructions must be added inside the initialization routine:

```
if (SYSCON.XPEN && (XPERCON & 0x07D3))
then { XPEREMU = XPERCON }
```

Of course, XPEREMU must be programmed after XPERCON and after SYSCON; in this way the final configuration for X-Peripherals is stored in XPEREMU and used for the emulation hardware setup.

XPEREMU (EB7Eh) XBUS Reset value xxxxh:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	XMISC EN	XI2C EN	XSSC EN	XASC EN	XPWM EN	XFLAS HEN	XRTC EN	XRAM2 EN	XRAM1 EN	CAN2 EN	CAN1 EN
-	-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Note: The bit meaning is exactly the same as in XPERCON.

22.13 Emulation dedicated registers

Four additional registers are implemented for emulation purposes only. Similarly to XPEREMU, they are write-only registers.

XEMU0 (EB76h) XBUS Reset value: xxxxh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XEMU0(15:0)															
W															

XEMU1 (EB78h) XBUS Reset value: xxxxh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XEMU1(15:0)															
W															

XEMU2 (EB7Ah) XBUS Reset value: xxxxh:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XEMU2(15:0)															
W															

XEMU3 (EB7Ch) XBUS Reset value: xxxxh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XEMU3(15:0)															
W															

Figure 57. Multiplexed bus with/without R/W delay and normal ALE

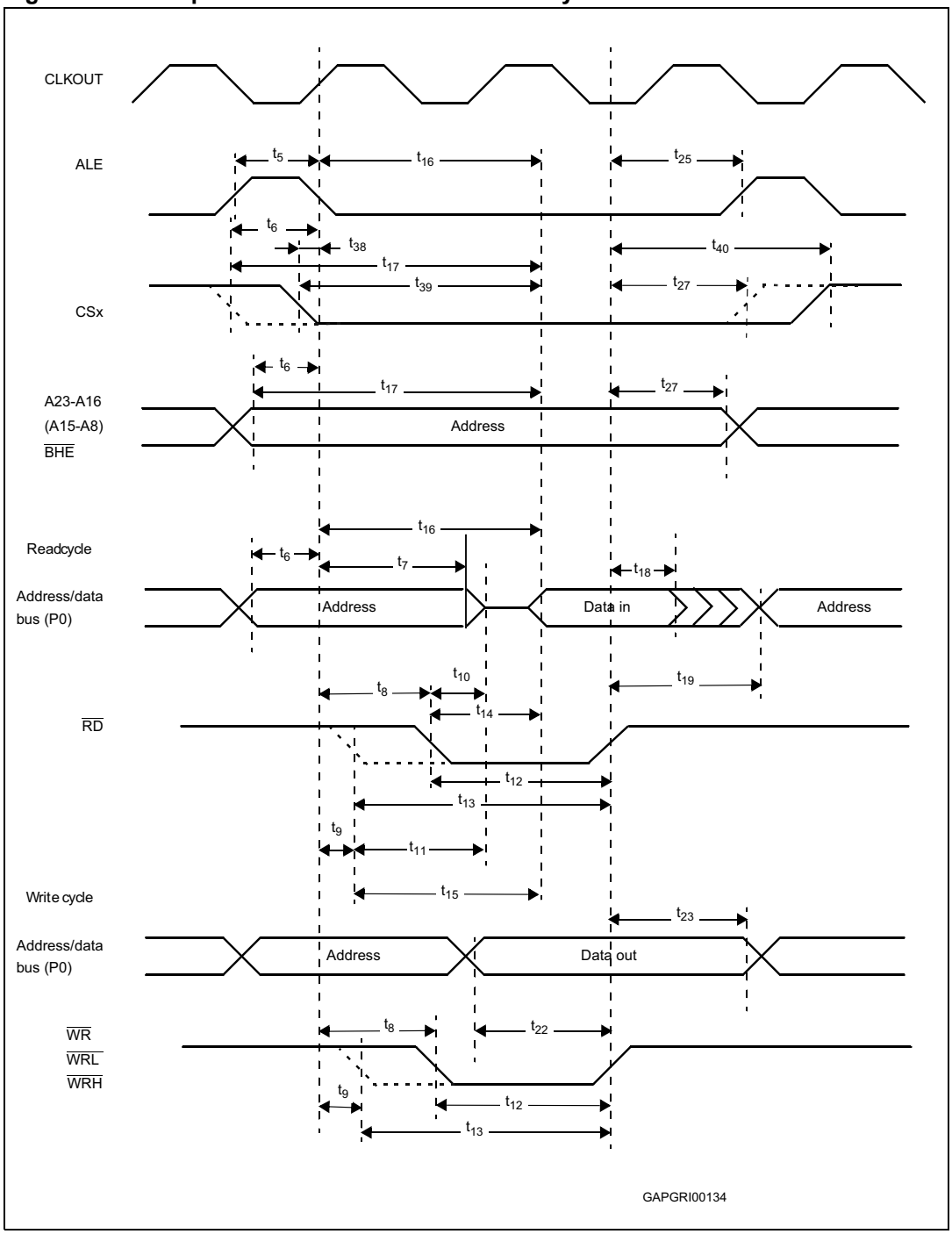
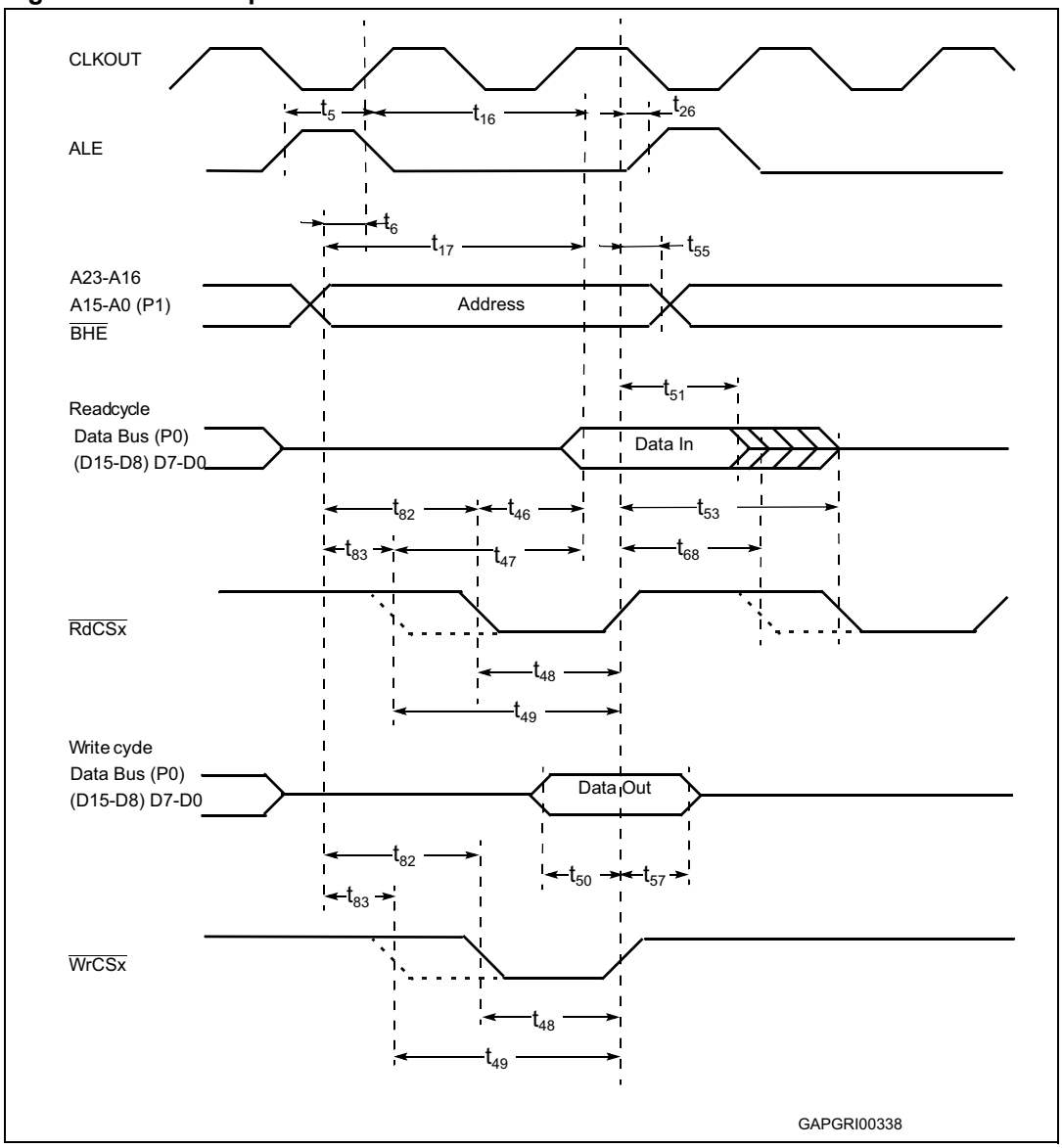


Figure 63. Demultiplexed bus with ALE and R/W CS



23.8.20 CLKOUT and $\overline{\text{READY}}$

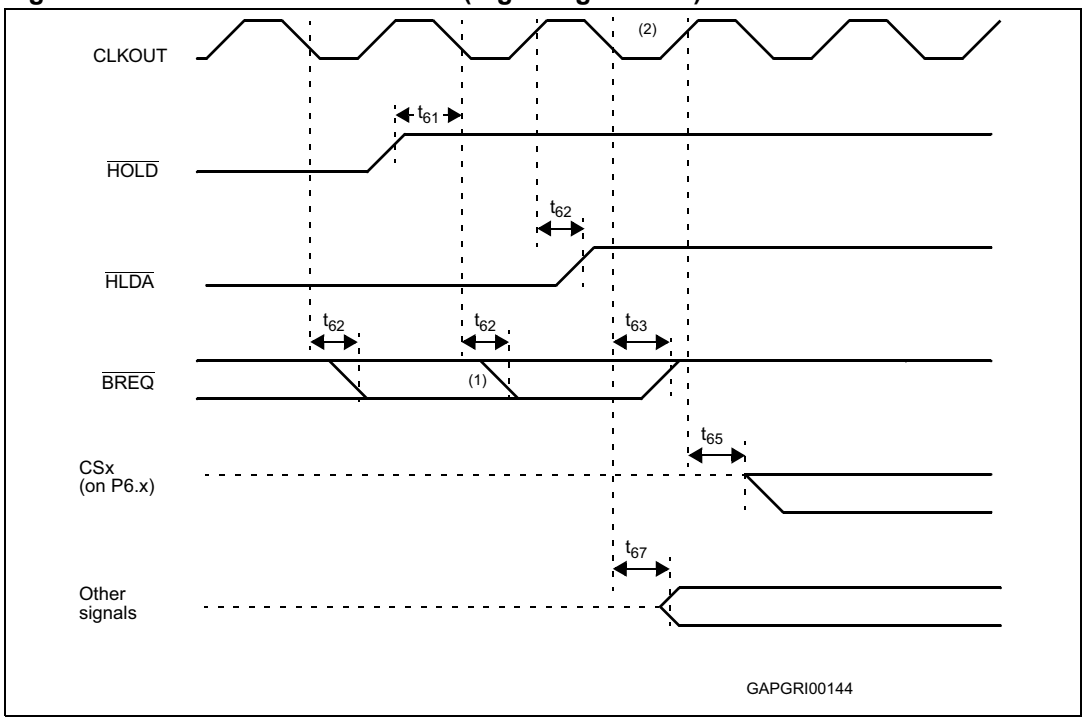
V_{DD} = 5V ±10%, V_{SS} = 0V, T_A = -40 to + 125°C, C_L = 50pF

Table 106. CLKOUT and $\overline{\text{READY}}$

Symbol	Parameter	f_{CPU} = 40 MHz TCL = 12.5ns		Variable CPU clock 1/2 TCL = 1 to 64 MHz		Unit
		Min.	Max.	Min.	Max.	
t_{29} **CC**	CLKOUT cycle time	25	25	2TCL	2TCL	ns
t_{30} **CC**	CLKOUT high time	9	-	TCL - 3.5	-	ns
t_{31} **CC**	CLKOUT low time	10		TCL - 2.5		ns
t_{32} **CC**	CLKOUT rise time	-	4	-	4	ns
t_{33} **CC**	CLKOUT fall time					ns
t_{34} **CC**	CLKOUT rising edge to ALE falling edge	- 2 + t_A	8 + t_A	- 2 + t_A	8 + t_A	ns
t_{35} **SR**	Synchronous $\overline{\text{READY}}$ setup time to CLKOUT	17	-	17	-	ns
t_{36} **SR**	Synchronous $\overline{\text{READY}}$ hold time after CLKOUT	2		2		ns
t_{37} **SR**	Asynchronous $\overline{\text{READY}}$ low time	35		2TCL + 10		ns
t_{58} **SR**	Asynchronous $\overline{\text{READY}}$ setup time(1)	17		17		ns
t_{59} **SR**	Asynchronous $\overline{\text{READY}}$ hold time(1)	2		2		ns
t_{60} **SR**	Async. $\overline{\text{READY}}$ hold time after $\overline{\text{RD}}$, $\overline{\text{WR}}$ high (Demultiplexed Bus)(2)	0	$2t_A + t_C + t_F$	0	$2t_A + t_C + t_F$	ns

1. These timings are given for characterization purposes only, in order to assure recognition at a specific clock edge.
2. Demultiplexed bus is the worst case. For multiplexed bus 2TCLs must be added to the maximum values. This adds even more time for deactivating READY. $2t_A$ and t_C refer to the next following bus cycle and t_F refers to the current bus cycle.

Figure 67. External bus arbitration (regaining the bus)



1. This is the last chance for $\overline{\text{BREQ}}$ to trigger the indicated regain-sequence. Even if $\overline{\text{BREQ}}$ is activated earlier, the regain-sequence is initiated by $\overline{\text{HOLD}}$ going high. Please note that $\overline{\text{HOLD}}$ may also be deactivated without the ST10F276E requesting the bus.
2. The next ST10F276E driven bus cycle may start here.