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	CANbus, EBI/EMI, SSC, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	111
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	12K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	144-BQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st10f269z2t6

Symbol	Pin	Type	Function
P0L.0 - P0L.7, P0H.0 P0H.1 - P0H.7	100-107, 108, 111-117	I/O	Two 8-bit bidirectional I/O ports P0L and P0H, bit-wise programmable for input or output via direction bit. Programming an I/O pin as input forces the corresponding output driver to high impedance state. In case of an external bus configuration, PORT0 serves as the address (A) and as the address / data (AD) bus in multiplexed bus modes and as the data (D) bus in demultiplexed bus modes. **Demultiplexed bus modes** Data Path Width: 8-bit 16-bit P0L.0 – P0L.7: D0 – D7 D0 - D7 P0H.0 – P0H.7 I/O D8 - D15 **Multiplexed bus modes** Data Path Width: 8-bit 16-bit P0L.0 – P0L.7: AD0 – AD7 AD0 - AD7 P0H.0 – P0H.7 A8 – A15 AD8 - AD15
P1L.0 - P1L.7 P1H.0 - P1H.7	118-125 128-135	I/O	Two 8-bit bidirectional I/O ports P1L and P1H, bit-wise programmable for input or output via direction bit. Programming an I/O pin as input forces the corresponding output driver to high impedance state. PORT1 is used as the 16-bit address bus (A) in demultiplexed bus modes and also after switching from a demultiplexed bus mode to a multiplexed bus mode. The following PORT1 pins have alternate functions:
	132	I	P1H.4 CC24IO CAPCOM2: CC24 Capture Input
	133	I	P1H.5 CC25IO CAPCOM2: CC25 Capture Input
	134	I	P1H.6 CC26IO CAPCOM2: CC26 Capture Input
	135	I	P1H.7 CC27IO CAPCOM2: CC27 Capture Input
XTAL1	138	I	XTAL1 Oscillator amplifier and/or external clock input.
XTAL2	137	O	XTAL2 Oscillator amplifier circuit output. To clock the device from an external source, drive XTAL1 while leaving XTAL2 unconnected. Minimum and maximum high / low and rise / fall times specified in the AC Characteristics must be observed.
$\overline{\text{RSTIN}}$	140	I	Reset Input with Schmitt-Trigger characteristics. A low level at this pin for a specified duration while the oscillator is running resets the ST10F269. An internal pull-up resistor permits power-on reset using only a capacitor connected to V_{SS}. In bidirectional reset mode (enabled by setting bit BDRSTEN in SYSCON register), the $\overline{\text{RSTIN}}$ line is pulled low for the duration of the internal reset sequence.
$\overline{\text{RSTOUT}}$	141	O	Internal Reset Indication Output. This pin is driven to a low level during hardware, software or watchdog timer reset. $\overline{\text{RSTOUT}}$ remains low until the EINIT (end of initialization) instruction is executed.
$\overline{\text{NMI}}$	142	I	Non-Maskable Interrupt Input. A high to low transition at this pin causes the CPU to vector to the NMI trap routine. If bit PWDCFG = '0' in SYSCON register, when the PWRDN (power down) instruction is executed, the $\overline{\text{NMI}}$ pin must be low in order to force the ST10F269 to go into power down mode. If $\overline{\text{NMI}}$ is high and PWDCFG ='0', when PWRDN is executed, the part will continue to run in normal mode. If not used, pin $\overline{\text{NMI}}$ should be pulled high externally.
V_{AREF}	37	-	A/D converter reference voltage.
V_{AGND}	38	-	A/D converter reference ground.
RPD	84	-	Timing pin for the return from interruptible powerdown mode and synchronous / asynchronous reset selection.

XPERCON (F024h / 12h) ESFR Reset Value: - - 05h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	-	-	RTCEN	XRAM2EN	XRAM1EN	CAN2EN	CAN1EN
											RW	RW	RW	RW	RW

CAN1EN	**CAN1 Enable Bit** '0': Accesses to the on-chip CAN1 XPeripheral and its functions are disabled. P4.5 and P4.6 pins can be used as general purpose I/Os. Address range 00'EF00h-00'EFFFh is only directed to external memory if CAN2EN is also '0'. '1': The on-chip CAN1 XPeripheral is enabled and can be accessed.
CAN2EN	**CAN2 Enable Bit** '0': Accesses to the on-chip CAN2 XPeripheral and its functions are disabled. P4.4 and P4.7 pins can be used as general purpose I/Os. Address range 00'EE00h-00'EEFFh is only directed to external memory if CAN1EN is also '0'. '1': The on-chip CAN2 XPeripheral is enabled and can be accessed.
XRAM1EN	**XRAM1 Enable Bit** '0': Accesses to external memory within space 00'E000h to 00'E7FFh. The 2K Bytes of internal XRAM1 are disabled. '1': Accesses to the internal 2K Bytes of XRAM1.
XRAM2EN	**XRAM2 Enable Bit** '0': Accesses to the external memory within space 00'C000h to 00'DFFFh. The 8K Bytes of internal XRAM2 are disabled. '1': Accesses to the internal 8K Bytes of XRAM2.
RTCEN	**RTC Enable Bit** '0': Accesses to the on-chip Real Time Clock are disabled, external access is performed. Address range 00'EC00h-00'ECFFh is only directed to external memory if CAN1EN and CAN2EN are '0' also '1': The on-chip Real Time Clock is enabled and can be accessed.

Note: - When both CAN are disabled via XPERCON setting, then any access in the address range 00'EE00h - 00'EFFFh will be directed to external memory interface, using the BUSCONx register corresponding to address matching ADDRSELx register. P4.4 and P4.7 can be used as General Purpose I/O when CAN2 is disabled, and P4.5 and P4.6 can be used as General Purpose I/O when CAN1 is disabled.

- The default XPER selection after Reset is identical to XBUS configuration of ST10C167: XCAN1 is enabled, XCAN2 is disabled, XRAM1 (2K Byte compatible XRAM) is enabled, XRAM2 (new 8K Byte XRAM) is disabled.

- Register XPERCON cannot be changed after the global enabling of XPeripherals, i.e. after the setting of bit XPEN in the SYSCON register.

- In EMUlation mode, all the XPERipherals are enabled (XPERCON bit are all set). The access to external memory and/or XBus is controlled by the bondout chip.

- When the Real Time Clock is disabled (RTCEN = 0), the clock oscillator is switch-off if the ST10 enters in power-down mode. Otherwise, when the Real Time Clock is enabled, the bit RTCOFF of the RTCCON register allows to choose the power-down mode of the clock oscillator (See Chapter : *Real Time Clock* on page 105).

8. MEM = any address inside the Flash memory space. Absolute addressing mode must be used (MOV MEM, Rn), and instruction must be executed from Flash memory space.

9. Odd word address = 4n-2 where n = 0, 1, 2, 3..., ex. 0002h, 0006h...

– Generally, command sequences cannot be written to Flash by instructions fetched from the Flash itself. Thus, the Flash commands must be written by instructions, executed from internal RAM or external memory.
– Command cycles on the CPU interface need not to be consecutively received (pauses allowed). The CPU interface delivers dummy read data for not used cycles within command sequences.
– All addresses of command cycles shall be defined only with **Register-indirect** addressing mode in the according move instructions. Direct addressing is not allowed for command sequences. Address segment or data page pointer are taken into account for the command address value.

5.3.7 - Reset Processing and Initial State

The Flash module distinguishes two kinds of CPU reset types

The lengthening of CPU reset:

– Is not reported to external devices by bidirectional pin
– Is not enabled in case of external start of CPU after reset.

5.4 - Flash Memory Configuration

The default memory configuration of the ST10F269 Memory is determined by the state of the $\overline{EA}$ pin at reset. This value is stored in the Internal ROM Enable bit (named ROMEN) of the SYSCON register.

When ROMEN = 0, the internal Flash is disabled and external ROM is used for startup control. Flash memory can later be enabled by setting the ROMEN bit of SYSCON to 1. The code performing this setting must not run from a segment of the external ROM to be replaced by a segment of the Flash memory, otherwise unexpected behaviour may occur.

For example, if external ROM code is located in the first 32K Bytes of segment 0, the first 32K Bytes of the Flash must then be enabled in segment 1. This is done by setting the ROMS1 bit of SYSCON to 0 before or simultaneously with setting of ROMEN bit. This must be done in the externally supplied program before the execution of the EINIT instruction.

If program execution starts from external memory, but access to the Flash memory mapped in segment 0 is later required, then the code that performs the setting of ROMEN bit must be executed either in the segment 0 but above address 00’8000h, or from the internal RAM.

Bit ROMS1 only affects the mapping of the first 32K Bytes of the Flash memory. All other parts of the Flash memory (addresses 01’8000h - 04’FFFFh) remain unaffected.

The SGTDIS Segmentation Disable / Enable must also be set to 0 to allow the use of the full 256K Bytes of on-chip memory in addition to the external boot memory. The correct procedure on changing the segmentation registers must also be observed to prevent an unwanted trap condition:

– Instructions that configure the internal memory must only be executed from external memory or from the internal RAM.
– An Absolute Inter-Segment Jump (JMPS) instruction must be executed after Flash enabling, to the next instruction, even if this next instruction is located in the consecutive address.
– Whenever the internal Memory is disabled, enabled or remapped, the DPPs must be explicitly (re)loaded to enable correct data accesses to the internal memory and/or external memory.

5.5 - Application Examples

5.5.1 - Handling of Flash Addresses

All command, Block, Data and register addresses to the Flash have to be located within the active Flash memory space. The active space is that address range to which the physical Flash addresses are mapped as defined by the user. When using data page pointer (DPP) for block addresses make sure that address bit A15 and A14 of the block address are reflected in both LSBs of the selected DPPS.

Note: - For Command Instructions, address bit A14, A15, A16 and A17 are don’t care. This simplify a lot the application software, because it minimize the use of DPP registers when using Command in the Command Interface.

- Direct addressing is not allowed for Command sequence operations to the Flash. Only Register-indirect addressing can be used for command, block or write-data accesses.

. . . .

5.6 - Bootstrap Loader

The built-in bootstrap loader (BSL) of the ST10F269 provides a mechanism to load the startup program through the serial interface after reset. In this case, no external memory or internal Flash memory is required for the initialization code starting at location 00'0000h (see Figure 5).

The bootstrap loader moves code/data into the internal RAM, but can also transfer data via the serial interface into an external RAM using a second level loader routine. Flash Memory (internal or external) is not necessary, but it may be used to provide lookup tables or "core-code" like a set of general purpose subroutines for I/O operations, number crunching, system initialization, etc.

The bootstrap loader can be used to load the complete application software into ROMless systems, to load temporary software into complete systems for testing or calibration, or to load a programming routine for Flash devices.

The BSL mechanism can be used for standard system startup as well as for special occasions like system maintenance (firmer update) or end-of-line programming or testing.

5.6.1 - Entering the Bootstrap Loader

The ST10F269 enters BSL mode when pin P0L.4 is sampled low at the end of a hardware reset. In this case the built-in bootstrap loader is activated independent of the selected bus mode.

The bootstrap loader code is stored in a special Boot-ROM. No part of the standard mask Memory or Flash Memory area is required for this.

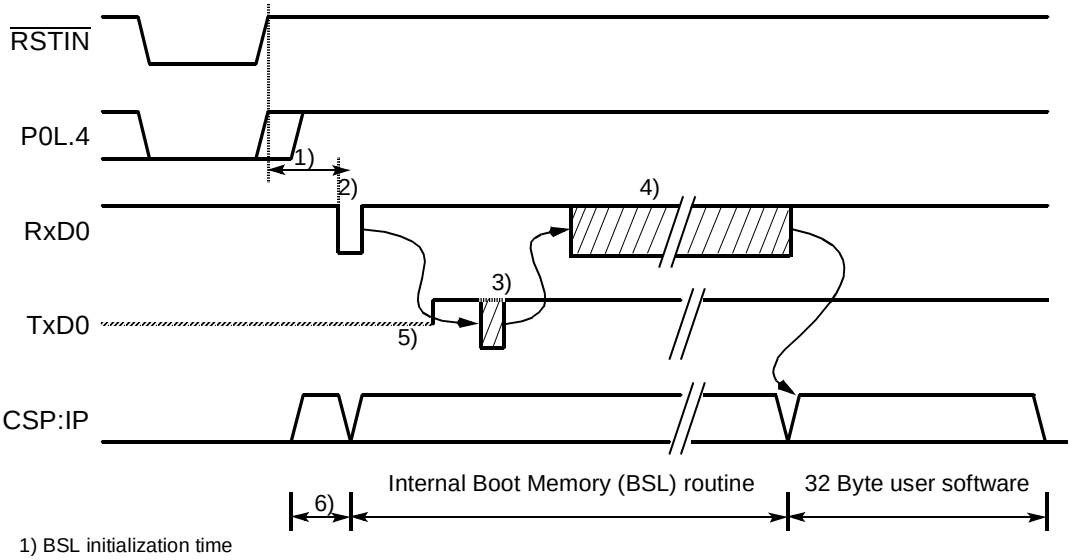
After entering BSL mode and the respective initialization the ST10F269 scans the RXD0 line to receive a zero Byte, one start bit, eight '0' data bits and one stop bit.

From the duration of this zero Byte it calculates the corresponding Baud rate factor with respect to the current CPU clock, initializes the serial interface ASC0 accordingly and switches pin TxD0 to output.

Using this Baud rate, an identification Byte is returned to the host that provides the loaded data.

This identification Byte identifies the device to be booted. The identification byte is D5h for ST10F269.

Figure 5 : Bootstrap Loader Sequence



1) BSL initialization time

2) Zero Byte (1 start bit, eight '0' data bits, 1 stop bit), sent by host.

3) Identification Byte (D5h), sent by ST10F269.

4) 32 Bytes of code / data, sent by host.

5) Caution: TxD0 is only driven a certain time after reception of the zero Byte.

6) Internal Boot ROM.

When the ST10F269 has entered BSL mode, the following configuration is automatically set (values that deviate from the normal reset values, are ***marked***):

Watchdog Timer:	***Disabled***	Register SYSCON:	0E00h
Context Pointer CP:	FA00h	Register STKUN:	FA40h
Stack Pointer SP:	FA40h	Register STKOV:	FA0Ch 0<->C
Register S0CON:	***8011h***	Register BUSCON0:	acc. to startup configuration
Register S0BG:	Acc. to '00' Byte	P3.10 / TXD0:	'**1**'
		DP3.10:	'**1**'

In this case, the watchdog timer is disabled, so the bootstrap loading sequence is not time limited.

Pin TXD0 is configured as output, so the ST10F269 can return the identification Byte.

Even if the internal Flash is enabled, no code can be executed out of it.

The hardware that activates the BSL during reset may be a simple pull-down resistor on P0L.4 for systems that use this feature upon every hardware reset.

A switchable solution (via jumper or an external signal) can be used for systems that only temporarily use the bootstrap loader (see Figure 6).

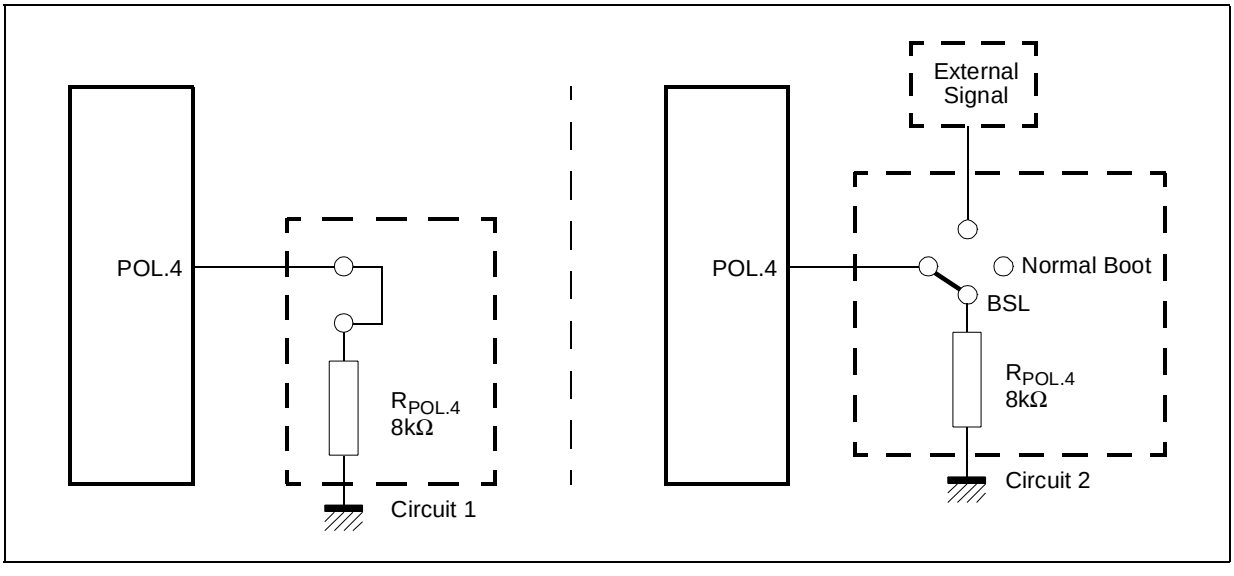
After sending the identification Byte the ASC0 receiver is enabled and is ready to receive the initial 32 Bytes from the host. A half duplex connection is therefore sufficient to feed the BSL.

5.6.2 - Memory Configuration After Reset

The configuration (and the accessibility) of the ST10F269's memory areas after reset in Bootstrap-Loader mode differs from the standard case. Pin $\overline{EA}$ is not evaluated when BSL mode is selected, and accesses to the internal Flash area are partly redirected, while the ST10F269 is in BSL mode (see Figure 7). All code fetches are made from the special Boot-ROM, while data accesses read from the internal user Flash. Data accesses will return undefined values on ROMless devices.

The code in the Boot-ROM is not an invariant feature of the ST10F269. User software should not try to execute code from the internal Flash area while the BSL mode is still active, as these fetches will be redirected to the Boot-ROM. The Boot-ROM will also "move" to segment 1, when the internal Flash area is mapped to segment 1 (see Figure 7).

Figure 6 : Hardware Provisions to Activate the BSL



The Table 5 shows the various combinations of pointer post-modification for each of these 2 new addressing modes. In this document the symbols "[$Rw_n\otimes$]" and "[$IDX_i\otimes$]" refer to these addressing modes.

Table 5 : Pointer Post-modification Combinations for IDXi and Rwn

Symbol	Mnemonic	Address Pointer Operation	
"[$IDX_i\otimes$]" stands for	[IDX_i]	(IDX_i) ← (IDX_i)	(no-op)
	[IDX_i+]	(IDX_i) ← (IDX_i) + 2	(i=0,1)
	[IDX_i-]	(IDX_i) ← (IDX_i) - 2	(i=0,1)
	[IDX_i + QX_j]	(IDX_i) ← (IDX_i) + (QX_j)	(i, j =0,1)
	[IDX_i - QX_j]	(IDX_i) ← (IDX_i) - (QX_j)	(i, j =0,1)
"[$Rw_n\otimes$]" stands for	[Rwn]	(Rwn) ← (Rwn)	(no-op)
	[Rwn+]	(Rwn) ← (Rwn) + 2	(n=0-15)
	[Rwn-]	(Rwn) ← (Rwn) - 2	(n=0-15)
	[Rwn + QR_j]	(Rwn) ← (Rwn) + (QR_j)	(n=0-15; j =0,1)
	[Rwn - QR_j]	(Rwn) ← (Rwn) - (QR_j)	(n=0-15; j =0,1)

Table 6 : MAC Registers Referenced as 'CoReg'

Registers	Description	Address in Opcode
MSW	MAC-Unit Status Word	00000b
MAH	MAC-Unit Accumulator High	00001b
MAS	"limited" MAH /signed	00010b
MAL	MAC-Unit Accumulator Low	00100b
MCW	MAC-Unit Control Word	00101b
MRW	MAC-Unit Repeat Word	00110b

Table 9 : Compare Modes

Compare Modes	Function
Mode 0	Interrupt-only compare mode; several compare interrupts per timer period are possible
Mode 1	Pin toggles on each compare match; several compare events per timer period are possible
Mode 2	Interrupt-only compare mode; only one compare interrupt per timer period is generated
Mode 3	Pin set '1' on match; pin reset '0' on compare time overflow; only one compare event per timer period is generated
Double Register Mode	Two registers operate on one pin; pin toggles on each compare match; several compare events per timer period are possible.

Table 10 : CAPCOM Timer Input Frequencies, Resolution and Periods (PQFP144 devices)

f_{CPU} = 40MHz	Timer Input Selection TxI							
	000b	001b	010b	011b	100b	101b	110b	111b
Pre-scaler for f_{CPU}	8	16	32	64	128	256	512	1024
Input Frequency	5MHz	2.5MHz	1.25MHz	625kHz	312.5kHz	156.25kHz	78.125kHz	39.1kHz
Resolution	200ns	400ns	0.8µs	1.6µs	3.2µs	6.4µs	12.8µs	25.6µs
Period	13.1ms	26.2ms	52.4ms	104.8ms	209.7ms	419.4ms	838.9ms	1.678s

Table 11 : CAPCOM Timer Input Frequencies, Resolution and Periods (TQFP144 devices)

f_{CPU} = 32MHz	Timer Input Selection TxI							
	000b	001b	010b	011b	100b	101b	110b	111b
Pre-scaler for f_{CPU}	8	16	32	64	128	256	512	1024
Input Frequency	4MHz	2MHz	1MHz	500KHz	250KHz	125KHz	62.5KHz	31.125KHz
Resolution	250ns	500ns	1µs	2µs	4µs	8µs	16µs	32µs
Period	16.4ms	32.8ms	65.5ms	131ms	262.1ms	524.3ms	1.05s	2.1s

Resolution and Period (TQFP144 devices) list the timer input frequencies, resolution and periods for each pre-scaler option at 40MHz (or 32MHz) CPU clock. This also applies to the Gated Timer Mode of T6 and to the auxiliary timer T5 in Timer and Gated Timer Mode.

Table 14 : GPT2 Timer Input Frequencies, Resolution and Period (PQFP144 devices)

f_{CPU} = 40MHz	Timer Input Selection T5I / T6I							
	000b	**001b**	**010b**	**011b**	**100b**	**101b**	**110b**	**111b**
Pre-scaler factor	4	8	16	32	64	128	256	512
Input Freq	10MHz	5MHz	2.5MHz	1.25MHz	625kHz	312.5kHz	156.25kHz	78.125kHz
Resolution	100ns	200ns	400ns	0.8µs	1.6µs	3.2µs	6.4µs	12.8µs
Period maximum	6.55ms	13.1ms	26.2ms	52.4ms	104.8ms	209.7ms	419.4ms	838.9ms

Table 15 : GPT2 Timer Input Frequencies, Resolution and Period (TQFP144 devices)

f_{CPU} = 32MHz	Timer Input Selection T5I / T6I							
	000b	**001b**	**010b**	**011b**	**100b**	**101b**	**110b**	**111b**
Pre-scaler factor	4	8	16	32	64	128	256	512
Input Freq	8MHz	4MHz	2MHz	1MHz	500KHz	250KHz	125KHz	62.5KHz
Resolution	125ns	250ns	500ns	1µs	2µs	4µs	8µs	16µs
Period maximum	8.19ms	16.4ms	32.8ms	65.5ms	131ms	262.1ms	524.3ms	1.05s

12.2.4 - Alternate Port Functions

Each port line has one associated programmable alternate input or output function.

– PORT0 and PORT1 may be used as address and data lines when accessing external memory.

– Port 2, Port 7 and Port 8 are associated with the capture inputs or compare outputs of the CAPCOM units and/or with the outputs of the PWM module.
Port 2 is also used for fast external interrupt inputs and for timer 7 input.

– Port 3 includes the alternate functions of timers, serial interfaces, the optional bus control signal $\overline{BHE}$ and the system clock output (CLKOUT).

– Port 4 outputs the additional segment address bit A16 to A23 in systems where segmentation is enabled to access more than 64K Bytes of memory.

– Port 5 is used as analog input channels of the A/D converter or as timer control signals.

– Port 6 provides optional bus arbitration signals ($\overline{BREQ}$, $\overline{HLDA}$, $\overline{HOLD}$) and chip select signals.

If an alternate output function of a pin is to be used, the direction of this pin must be programmed for output (DPx.y='1'), except for some signals that are used directly after reset and are configured automatically. Otherwise the pin remains in the high-impedance state and is not effected by the alternate output function. The respective port latch should hold a '1', because its output is ANDed with the alternate output data (except for PWM output signals).

If an alternate input function of a pin is used, the direction of the pin must be programmed for input (DPx.y='0') if an external device is driving the pin. The input direction is the default after reset. If no external device is connected to the pin, however, one can also set the direction for this pin to output.

In this case, the pin reflects the state of the port output latch. Thus, the alternate input function reads the value stored in the port output latch. This can be used for testing purposes to allow a software trigger of an alternate input function by writing to the port output latch.

On most of the port lines, the application software must set the proper direction when using an alternate input or output function of a pin. This is done by setting or clearing the direction control bit DPx.y of the pin before enabling the alternate function. There are port lines, however, where the direction of the port line is switched automatically.

For instance, in the multiplexed external bus modes of PORT0, the direction must be switched several times for an instruction fetch in order to output the addresses and to input the data. Obviously, this cannot be done through instructions. In these cases, the direction of the port line is switched automatically by hardware if the alternate function of such a pin is enabled.
To determine the appropriate level of the port output latches check how the alternate data output is combined with the respective port latch output.

There is one basic structure for all port lines supporting only one alternate input function. Port lines with only one alternate output function, however, have different structures. It has to be adapted to support the normal and the alternate function features.

All port lines that are not used for these alternate functions may be used as general purpose I/O lines. When using port pins for general purpose output, the initial output value should be written to the port latch prior to enabling the output drivers, in order to avoid undesired transitions on the output pins. This applies to single pins as well as to pin groups (see examples below).

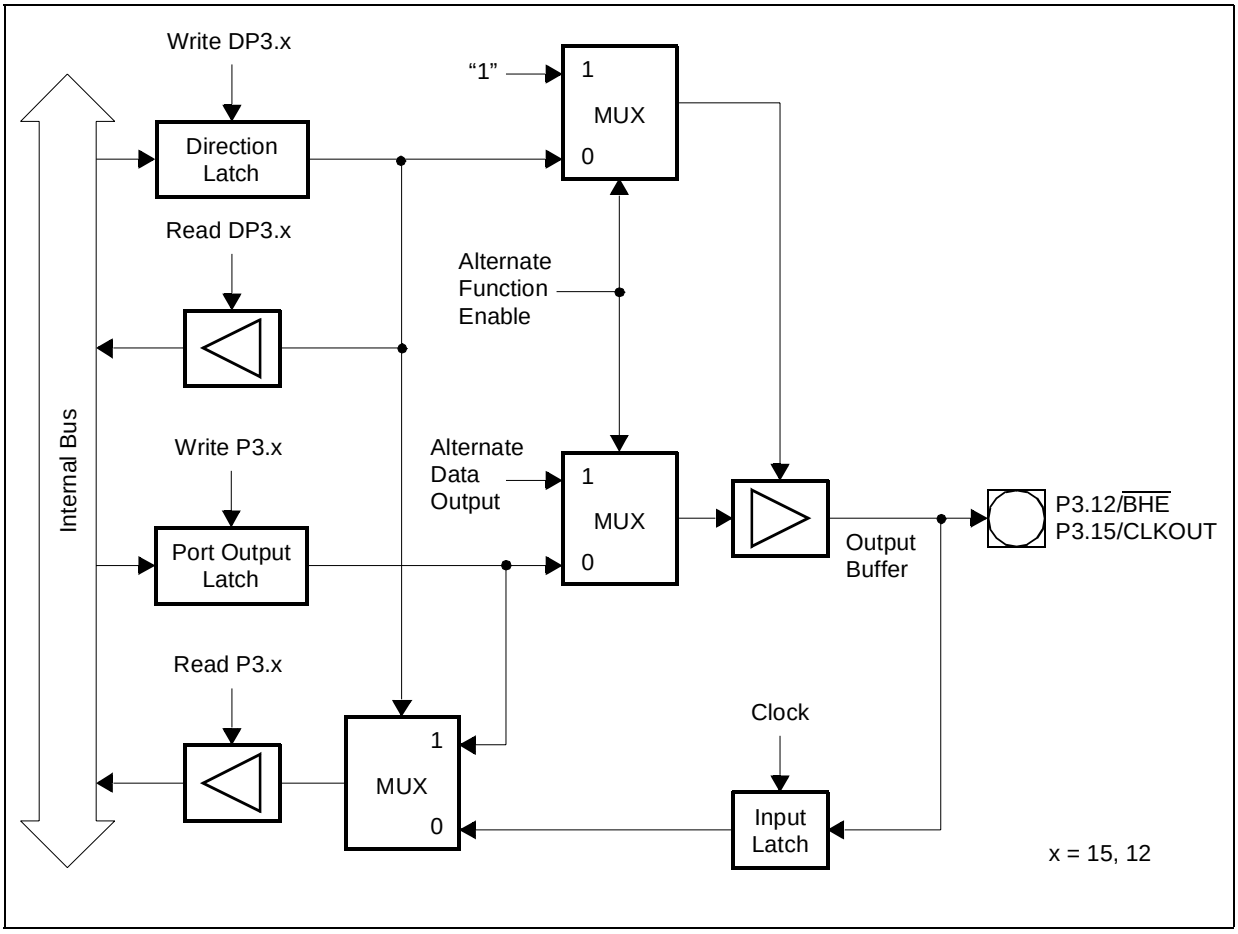
```
SINGLE_BIT: BSET     P4.7              ; Initial output level is "high"
            BSET     DP4.7             ; Switch on the output driver
BIT_GROUP:  BFLDH    P4, #24H, #24H    ; Initial output level is "high"
            BFLDH    DP4, #24H, #24H   ; Switch on the output drivers
```

Note: When using several BSET pairs to control more pins of one port, these pairs must be separated by instructions, which do not apply to the respective port (See Chapter : *Central Processing Unit (CPU)* on page 35).

Pin P3.12 ($\overline{BHE}$/$\overline{WRH}$) is another pin with an alternate output function, however, its structure is slightly different.

After reset the $\overline{BHE}$ or $\overline{WRH}$ function must be used depending on the system start-up configuration. In either of these cases, there is no possibility to program any port latches before. Thus, the appropriate alternate function is selected automatically. If $\overline{BHE}$/$\overline{WRH}$ is not used in the system, this pin can be used for general purpose I/O by disabling the alternate function (BYTDIS = '1' / WRCFG='0').

Figure 29 : Block Diagram of Pins P3.15 (CLKOUT) and P3.12 ($\overline{BHE}$/$\overline{WRH}$)



Note: Enabling the $\overline{BHE}$ or $\overline{WRH}$ function automatically enables the P3.12 output driver. Setting bit DP3.12='1' is not required.
During bus hold pin P3.12 is switched back to its standard function and is then controlled by DP3.12 and P3.12. Keep DP3.12 = '0' in this case to ensure floating in hold mode.

Figure 31 : Block Diagram of a Port 4 Pin

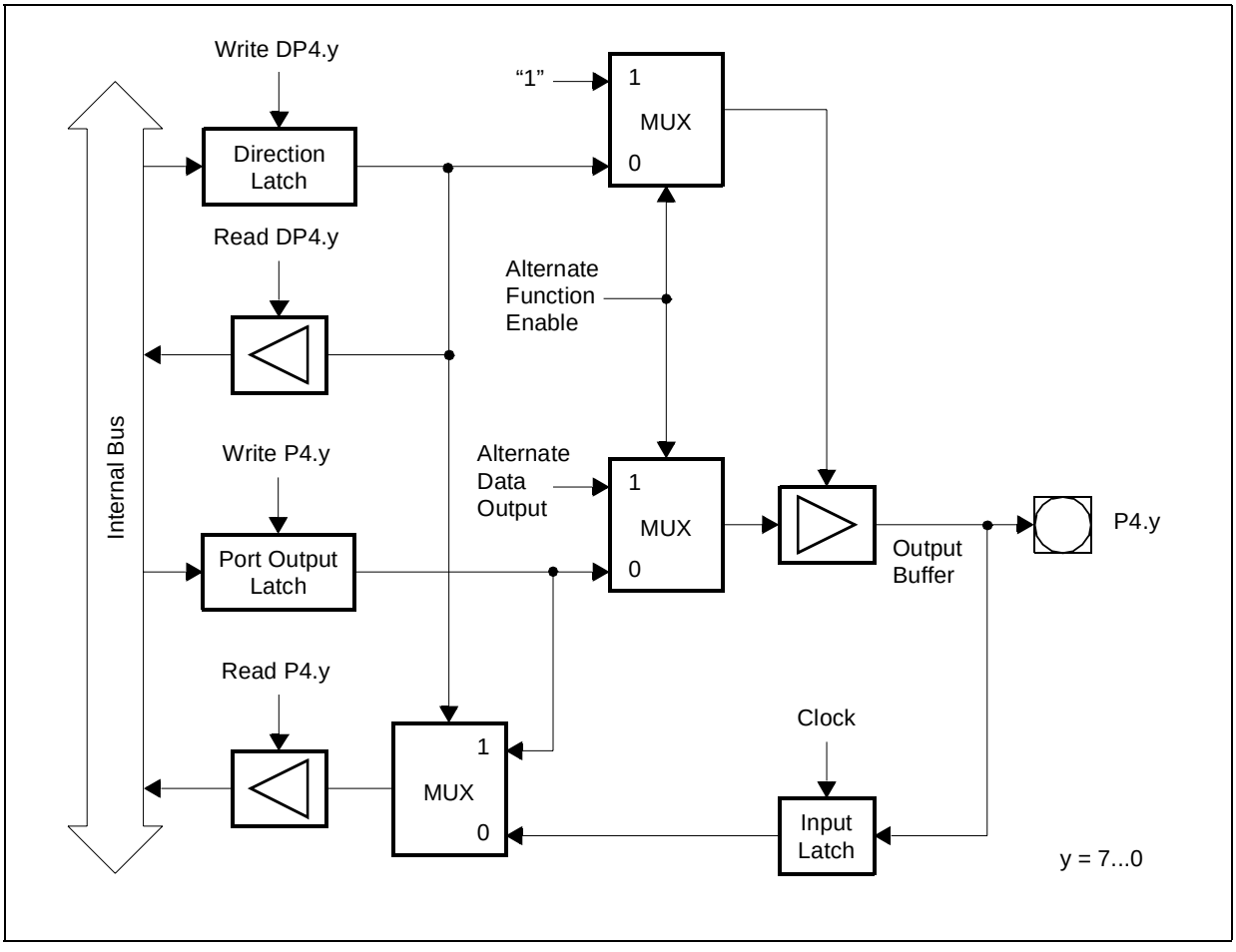


Table 29 : Commonly Used Baud Rates by Reload Value and Deviation Errors (TQFP144 devices)

S0BRS = '0', f_{CPU} = 32MHz			S0BRS = '1', f_{CPU} = 32MHz		
Baud Rate (Baud)	**Deviation Error**	**Reload Value**	**Baud Rate (Baud)**	**Deviation Error**	**Reload Value**
1000 000	±0.0%	0000h	666 667	±0.0%	0000h
56000	+5.0% / -0.8%	0010h / 001h	56000	+8.2% / -0.8%	000Ah / 000Bh
38400	+0.2% / -3.5%	0019h / 0020h	38400	+2.1% / -3.5%	0010h / 0011h
19200	+0.2% / -1.7%	0033h / 0034h	19200	+2.1% / -0.8%	0021h / 0022h
9600	+0.2% / -0.8%	0067h/ 0068h	9600	+0.6% / -0.8%	0044h / 0045h
4800	+0.5% / -0.3%	00CFh / 00CEh	4800	+0.6% / -0.1%	0089h / 008Ah
2400	+0.2% / -0.1%	019Fh / 01A0h	2400	+0.3% / -0.1%	0114h / 0115h
1200	+0.1% / -0.1%	0340h / 0341h	1200	+0.1% / -0.1%	022Ah / 022Bh
600	+0.1% / -0.1%	0681h / 0682h	600	+0.1% / -0.1%	0456h / 0457h
95	+0.1% / -0.1%	291Dh / 291Eh	75	+0.1% / 0.1%	22B7h / 22B8h
			63	+0.1% / -0.1%	2955h / 2956h

Note: The deviation errors given in the Table 29 are rounded. To avoid deviation errors use a Baud rate crystal (providing a multiple of the ASC0/SSC sampling frequency).

Baud Rate Generation

The Baud rate generator is clocked by $f_{CPU}/2$. The timer is counting downwards and can be started or stopped through the global enable bit SSCEN in register SSCCON. Register SSCBR is the dual-function Baud Rate Generator/Reload register. Reading SSCBR, while the SSC is enabled, returns the content of the timer. Reading SSCBR, while the SSC is disabled, returns the programmed reload value. In this mode the desired reload value can be written to SSCBR.

Note Never write to SSCBR, while the SSC is enabled.

The formulas below calculate the resulting Baud rate for a given reload value and the required reload value for a given Baud rate:

$$\text{Baud rate}_{SSC} = \frac{f_{CPU}}{2 \times [(SSCBR) + 1]}$$

$$SSCBR = \left(\frac{f_{CPU}}{2 \times \text{Baud rate}_{SSC}} \right) - 1$$

(SSCBR) represents the content of the reload register, taken as unsigned 16-bit integer.

Table 32 lists some possible Baud rates against the required reload values and the resulting bit times for a 40MHz CPU clock.

Table 32 : Synchronous Baud Rate and Reload Values (PQFP144 devices)

Baud Rate	Bit Time	Reload Value
Reserved use a reload value > 0.	---	---
10M Baud	100ns	0001h
5M Baud	200ns	0003h
2.5M Baud	400ns	0007h
1M Baud	1µs	0013h
100K Baud	10µs	00C7h
10K Baud	100µs	07CFh
1K Baud	1ms	4E1Fh
306 Baud	3.26ms	FF4Eh

Table 33 lists some possible Baud rates against the required reload values and the resulting bit times for a 32MHz CPU clock.

.

Table 33 : Synchronous Baud Rate and Reload Values (TQFP144 devices)

Baud Rate	Bit Time	Reload Value
Reserved use a reload value > 0.	---	---
8MBaud	125ns	0001h
4MBaud	250ns	0003h
2MBaud	500ns	0007h
1MBaud	1µs	000Fh
500KBaud	2µs	001Fh
100KBaud	10µs	009Fh
10KBaud	100µs	030Ch
1K Baud	1ms	3E7Fh
244.14 Baud	5.24ms	FFFFh

	'0': Pins $\overline{WR}$ and $\overline{BHE}$ retain their normal function '1': Pin $\overline{WR}$ acts as $\overline{WRL}$, pin $\overline{BHE}$ acts as $\overline{WRH}$.
CLKEN	**System Clock Output Enable (CLKOUT)** '0': CLKOUT disabled: pin may be used for general purpose I/O '1': CLKOUT enabled: pin outputs the system clock signal.
BYTDIS	**Disable/Enable Control for Pin $\overline{BHE}$ (Set according to data bus width)** '0': Pin $\overline{BHE}$ enabled '1': Pin $\overline{BHE}$ disabled, pin may be used for general purpose I/O.
ROMEN	**Internal Memory Enable (Set according to pin $\overline{EA}$ during reset)** '0': Internal Memory disabled: accesses to the Memory area use the external bus '1': Internal Memory enabled.
SGTDIS	**Segmentation Disable/Enable Control** '0': Segmentation enabled (CSP is saved/restored during interrupt entry/exit) '1': Segmentation disabled (Only IP is saved/restored).
ROMS1	**Internal Memory Mapping** '0': Internal Memory area mapped to segment 0 (00'0000H...00'7FFFH) '1': Internal Memory area mapped to segment 1 (01'0000H...01'7FFFH).
STKSZ	**System Stack Size** Selects the size of the system stack (in the internal RAM) from 32 to 1024 words.

BUSCON0 (FF0Ch / 86h) SFR Reset Value: 0xx0h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSWEN0	CSREN0	RDYPOL0	RDYEN0	-	**BUS ACT0**	**ALE CTL0**	-	BTYP		MTTC0	RWDC0	MCTC			
RW	RW	RW	RW		RW[2]	RW[2]		RW[1]		RW	RW	RW			

BUSCON1 (FF14h / 8Ah) SFR Reset Value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSWEN1	CSREN1	RDYPOL1	RDYEN1	-	BUSACT1	ALECTL1	-	BTYP		MTTC1	RWDC1	MCTC			
RW	RW	RW	RW		RW	RW		RW		RW	RW	RW			

BUSCON2 (FF16h / 8Bh) SFR Reset Value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSWEN2	CSREN2	RDYPOL2	RDYEN2	-	BUSACT2	ALECTL2	-	BTYP		MTTC2	RWDC2	MCTC			
RW	RW	RW	RW		RW	RW		RW		RW	RW	RW			

BUSCON3 (FF18h / 8Ch) SFR Reset Value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSWEN3	CSREN3	RDYPOL3	RDYEN3	-	BUSACT3	ALECTL3	-	BTYP		MTTC3	RWDC3	MCTC			
RW	RW	RW	RW		RW	RW		RW		RW	RW	RW			

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 [1 - 2]			R [2]		R [2]		R [2]

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)** 0 0: 3 $\overline{CS}$ lines: $\overline{CS}$2...$\overline{CS}$0 0 1: 2 $\overline{CS}$ lines: $\overline{CS}$1...$\overline{CS}$0 1 0: No $\overline{CS}$ line at all 1 1: 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)** 0 0: 4-bit segment address: A19...A16 0 1: No segment address lines at all 1 0: 8-bit segment address: A23...A16 1 1: 2-bit segment address: A17...A16 (Default without pull-downs)
CLKSEL [1 - 2]	**System Clock Selection** 000: $f_{CPU} = 2.5 \times f_{OSC}$ 001: $f_{CPU} = 0.5 \times f_{OSC}$ 010: $f_{CPU} = 1.5 \times f_{OSC}$ 011: $f_{CPU} = f_{OSC}$ 100: $f_{CPU} = 5 \times f_{OSC}$ 101: $f_{CPU} = 2 \times f_{OSC}$ 110: $f_{CPU} = 3 \times f_{OSC}$ 111: $f_{CPU} = 4 \times f_{OSC}$

Notes: 1. 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".

2. These bits are set according to Port 0 configuration during any reset sequence.

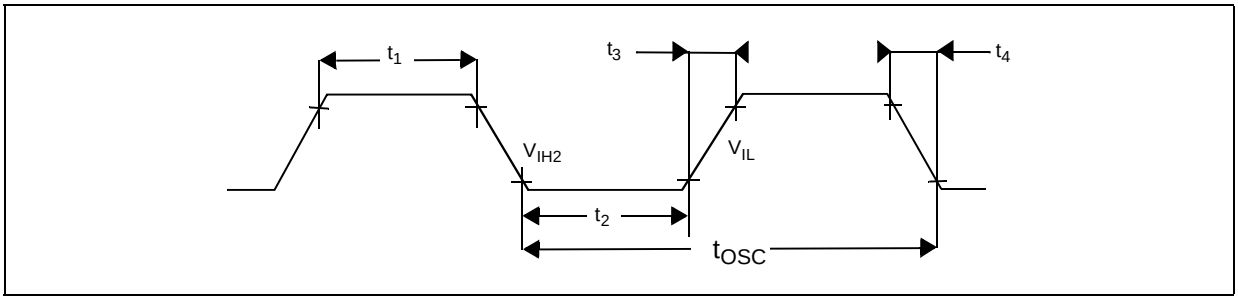
3. RP0H is a read only register.

V_{DD} = 5V ± 10%, V_{SS} = 0V, T_A = -40 to +125 °C (TQFP144 devices)

Parameter	Symbol		$f_{CPU} = f_{XTAL}$		$f_{CPU} = f_{XTAL} / 2$		$f_{CPU} = f_{XTAL} \times F$ F = 1.5/2,/2.5/3/4/5		Unit
			Minimum	Maximum	Minimum	Maximum	Minimum	Maximum	
Oscillator period	t_{OSC}	SR	31.25[1]	–	15.625	–	31.25 x N	–	ns
High time	t_1	SR	12.5[2]	–	6.25[2]	–	12.5[2]	–	ns
Low time	t_2	SR	12.5[2]	–	6.25[2]	–	12.5[2]	–	ns
Rise time	t_3	SR	–	3.125[2]	–	1.56[2]	–	3.125[2]	ns
Fall time	t_4	SR	–	3.125[2]	–	1.56[2]	–	3.125[2]	ns

1. Theoretical minimum. The real minimum value depends on the duty cycle of the input clock signal. 32MHz is the maximum input frequency when using an external crystal oscillator. However, 32MHz can be applied with an external clock source.
2. The input clock signal must reach the defined levels V_{IL} and V_{IH2}.

Figure 70 : External Clock Drive XTAL1



21.4.9 - Memory Cycle Variables

The tables below use three variables which are derived from the BUSCONx registers and represent the special characteristics of the programmed memory cycle. The following table describes, how these variables are to be computed.

Description	Symbol	Values
ALE Extension	t_A	TCL x [ALECTL]
Memory Cycle Time wait states	t_C	2TCL x (15 - [MCTC])
Memory Tri-state Time	t_F	2TCL x (1 - [MTTC])

21.4.10 - Multiplexed Bus

V_{DD} = 5V ± 10%, V_{SS} = 0V, T_A = -40 to +125°C, C_L = 50pF,
ALE cycle time = 6 TCL + $2t_A$ + t_C + t_F (75ns at 40MHz CPU clock without wait states, PQFP144 devices).

Table 46 : Multiplexed Bus Characteristics (PQFP144 devices)

Symbol		Parameter	Max. CPU Clock = 40MHz		Variable CPU Clock 1/2 TCL = 1 to 40MHz		Unit
			min.	max.	min.	max.	
t_5	CC	ALE high time	4 + t_A	–	TCL - 8.5 + t_A	–	ns
t_6	CC	Address setup to ALE	2 + t_A	–	TCL - 10.5 + t_A	–	ns
t_7	CC	Address hold after ALE 1	4 + t_A	–	TCL - 8.5 + t_A	–	ns
t_8	CC	ALE falling edge to $\overline{RD}$, $\overline{WR}$ (with RW-delay)	4 + t_A	–	TCL - 8.5 + t_A	–	ns
t_9	CC	ALE falling edge to $\overline{RD}$, $\overline{WR}$ (no RW-delay)	-8.5 + t_A	–	-8.5 + t_A	–	ns
t_{10}	CC	Address float after $\overline{RD}$, $\overline{WR}$ (with RW-delay) 1	–	6	–	6	ns
t_{11}	CC	Address float after $\overline{RD}$, $\overline{WR}$ (no RW-delay) 1	–	18.5	–	TCL + 6	ns
t_{12}	CC	$\overline{RD}$, $\overline{WR}$ low time (with RW-delay)	15.5 + t_C	–	2 TCL -9.5 + t_C	–	ns
t_{13}	CC	$\overline{RD}$, $\overline{WR}$ low time (no RW-delay)	28 + t_C	–	3 TCL -9.5 + t_C	–	ns
t_{14}	SR	$\overline{RD}$ to valid data in (with RW-delay)	–	6 + t_C	–	2 TCL - 19 + t_C	ns
t_{15}	SR	$\overline{RD}$ to valid data in (no RW-delay)	–	18.5 + t_C	–	3 TCL - 19 + t_C	ns
t_{16}	SR	ALE low to valid data in	–	18.5 + t_A + t_C	–	3 TCL - 19 + t_A + t_C	ns
t_{17}	SR	Address/Unlatched $\overline{CS}$ to valid data in	–	22 + $2t_A$ + t_C	–	4 TCL - 28 + $2t_A$ + t_C	ns
t_{18}	SR	Data hold after $\overline{RD}$ rising edge	0	–	0	–	ns
t_{19}	SR	Data float after $\overline{RD}$ 1	–	16.5 + t_F	–	2 TCL - 8.5 + t_F	ns
t_{22}	CC	Data valid to $\overline{WR}$	10 + t_C	–	2 TCL -15 + t_C	–	ns
t_{23}	CC	Data hold after $\overline{WR}$	4 + t_F	–	2 TCL - 8.5 + t_F	–	ns
t_{25}	CC	ALE rising edge after $\overline{RD}$, $\overline{WR}$	15 + t_F	–	2 TCL -10 + t_F	–	ns
t_{27}	CC	Address/Unlatched $\overline{CS}$ hold after $\overline{RD}$, $\overline{WR}$	10 + t_F	–	2 TCL -15 + t_F	–	ns
t_{38}	CC	ALE falling edge to Latched $\overline{CS}$	-4 - t_A	10 - t_A	-4 - t_A	10 - t_A	ns
t_{39}	SR	Latched $\overline{CS}$ low to Valid Data In	–	18.5 + t_C + $2t_A$	–	3 TCL - 19 + t_C + $2t_A$	ns
t_{40}	CC	Latched $\overline{CS}$ hold after $\overline{RD}$, $\overline{WR}$	27 + t_F	–	3 TCL - 10.5 + t_F	–	ns

Figure 74 : External Memory Cycle: Multiplexed Bus, With / Without Read / Write Delay, Extended ALE, Read / Write Chip Select

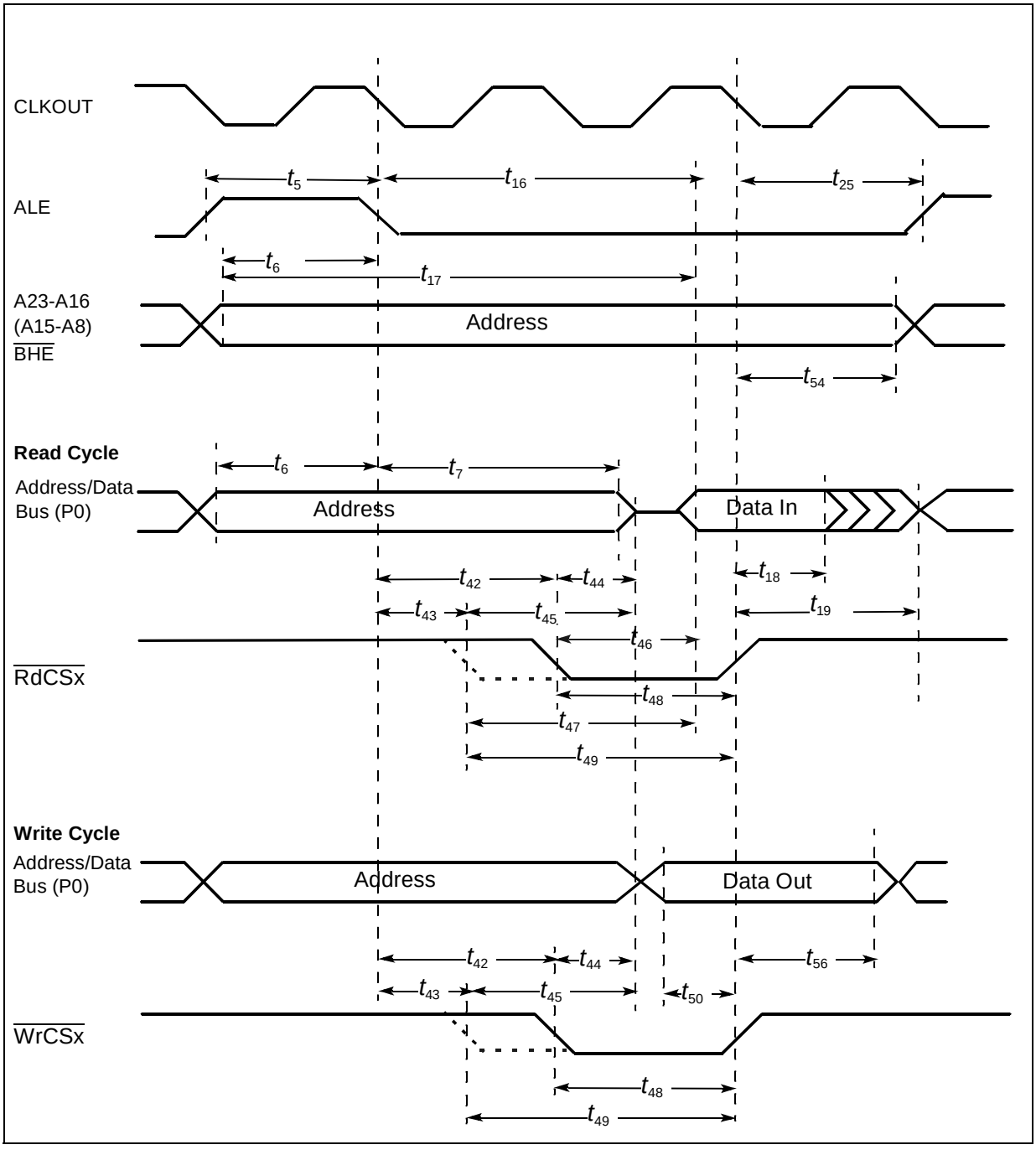
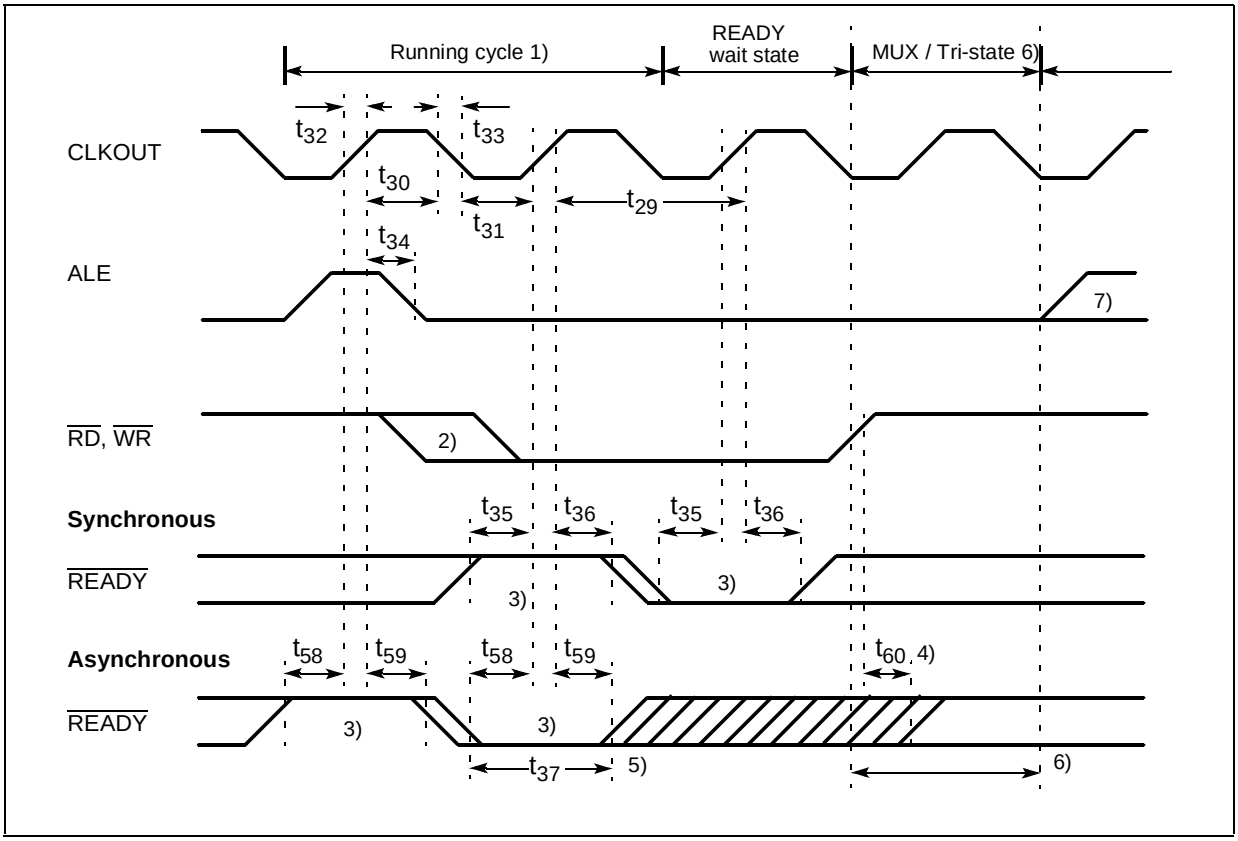


Figure 79 : CLKOUT and $\overline{\text{READY}}$



Notes: 1. Cycle as programmed, including MCTC wait states (Example shows 0 MCTC WS).

2. The leading edge of the respective command depends on RW-delay.

3. $\overline{\text{READY}}$ sampled HIGH at this sampling point generates a READY controlled wait state, $\overline{\text{READY}}$ sampled LOW at this sampling point terminates the currently running bus cycle.

4. $\overline{\text{READY}}$ may be deactivated in response to the trailing (rising) edge of the corresponding command ($\overline{\text{RD}}$ or $\overline{\text{WR}}$).

5. If the Asynchronous $\overline{\text{READY}}$ signal does not fulfill the indicated setup and hold times with respect to CLKOUT (e.g. because CLKOUT is not enabled), it must fulfill t_{37} in order to be safely synchronized. This is guaranteed, if READY is removed in response to the command (see Note 4)).

6. Multiplexed bus modes have a MUX wait state added after a bus cycle, and an additional MTTC wait state may be inserted here. For a multiplexed bus with MTTC wait state this delay is 2 CLKOUT cycles, for a demultiplexed bus without MTTC wait state this delay is zero.

7. The next external bus cycle may start here.

21.4.14 - High-Speed Synchronous Serial Interface (SSC) Timing

21.4.14.1 Master Mode

V_{CC} = 5V ±10%, V_{SS} = 0V, CPU clock = 40MHz, T_A = -40 to +125°C, C_L = 50pF (PQFP144 devices)

Symbol		Parameter	Maximum Baud rate = 10M Baud (<SSCBR> = 0001h)		Variable Baud rate (<SSCBR>=0001h-FFFFh)		Unit
			Minimum	Maximum	Minimum	Maximum	
t_{300}	CC	SSC clock cycle time	100	100	8 TCL	262144 TCL	ns
t_{301}	CC	SSC clock high time	40	–	t_{300}/2 - 10	–	ns
t_{302}	CC	SSC clock low time	40	–	t_{300}/2 - 10	–	ns
t_{303}	CC	SSC clock rise time	–	10	–	10	ns
t_{304}	CC	SSC clock fall time	–	10	–	10	ns
t_{305}	CC	Write data valid after shift edge	–	15	–	15	ns
t_{306}	CC	Write data hold after shift edge [1]	-2	–	-2	–	ns
t_{307p}	SR	Read data setup time before latch edge, phase error detection on (SSCPEN = 1)	37.5	–	2TCL+12.5	–	ns
t_{308p}	SR	Read data hold time after latch edge, phase error detection on (SSCPEN = 1)	50	–	4TCL	–	ns
t_{307}	SR	Read data setup time before latch edge, phase error detection off (SSCPEN = 0)	25	–	2TCL	–	ns
t_{308}	SR	Read data hold time after latch edge, phase error detection off (SSCPEN = 0)	0	–	0	–	ns

Note: 1. Timing guaranteed by design.

*The formula for SSC Clock Cycle time is: t_{300} = 4 TCL * (<SSCBR> + 1)*
Where <SSCBR> represents the content of the SSC Baud rate register, taken as unsigned 16-bit integer.

V_{CC} = 5V ±10%, V_{SS} = 0V, CPU clock = 32MHz, T_A = -40 to +125°C, C_L = 50pF (TQFP144 devices)

Symbol		Parameter	Maximum Baud rate=6.25MBd (<SSCBR> = 0001h)		Variable Baud rate (<SSCBR>=0001h-FFFFh)		Symbol
			Minimum	Maximum	Minimum	Maximum	
t_{310}	SR	SSC clock cycle time	125	–	8 TCL	262144 TCL	t_{310}
t_{311}	SR	SSC clock high time	52.5	–	t_{310}/2 - 10	–	t_{311}
t_{312}	SR	SSC clock low time	52.5	–	t_{310}/2 - 10	–	t_{312}
t_{313}	SR	SSC clock rise time	–	10	–	10	t_{313}
t_{314}	SR	SSC clock fall time	–	10	–	10	t_{314}
t_{315}	CC	Write data valid after shift edge	–	45.25	–	2 TCL + 14	t_{315}
t_{316}	CC	Write data hold after shift edge	0	–	0	–	t_{316}
t_{317p}	SR	Read data setup time before latch edge, phase error detection on (SSCPEN = 1)	78.125	–	4TCL + 15.625	–	t_{317p}
t_{318p} [1]	SR	Read data hold time after latch edge, phase error detection on (SSCPEN = 1)	109.375	–	6TCL + 15.625	–	t_{318p} [1]
t_{317}	SR	Read data setup time before latch edge, phase error detection off (SSCPEN = 0)	6	–	6	–	t_{317}