

## ST9 EXTERNAL MEMORY INTERFACE CONFIGURATION

### INTRODUCTION

This application note presents the different ST9 resources for configuring and initializing its external memory interface.

The ST9 has a single 4 Mbyte memory space segmented in 64 segments of 64 Kbytes, plus an independent register file space. The memory space contains internal memories (internal ROM and RAM with predefined addresses) and you can map your external memories (at the addresses in any segments not used for internal memories). Please refer to the MMU chapter of the ST9 datasheet for more information on the way this memory space is addressed.

The ST9 external memory access cycle is composed of 2 clock phases (cf. Figure 1):

- Phase T1: the memory address is output through the ST9 EMI (External Memory Interface).
- Phase T2: if the memory access is a Read cycle, the data signals are sampled by the ST9. If the memory access is a Write cycle, the ST9 outputs data to be written in external memory.

The different signals provided and used by the EMI are described in [Section 1](#) of this application note, and their configuration is explained in [Section 2.2](#). A software example of the ST9 EMI configuration is given in [Section 5](#).

### 1 SIGNAL DESCRIPTION

The External Memory Interface for the ST9 microcontroller exists in two models:

- a) small model that allows you to address a maximum of 64 Kbytes of external memory and
- b) large model which can address the full ST9 memory space (4 Mbytes).

#### 1.1 SMALL MEMORY MODEL

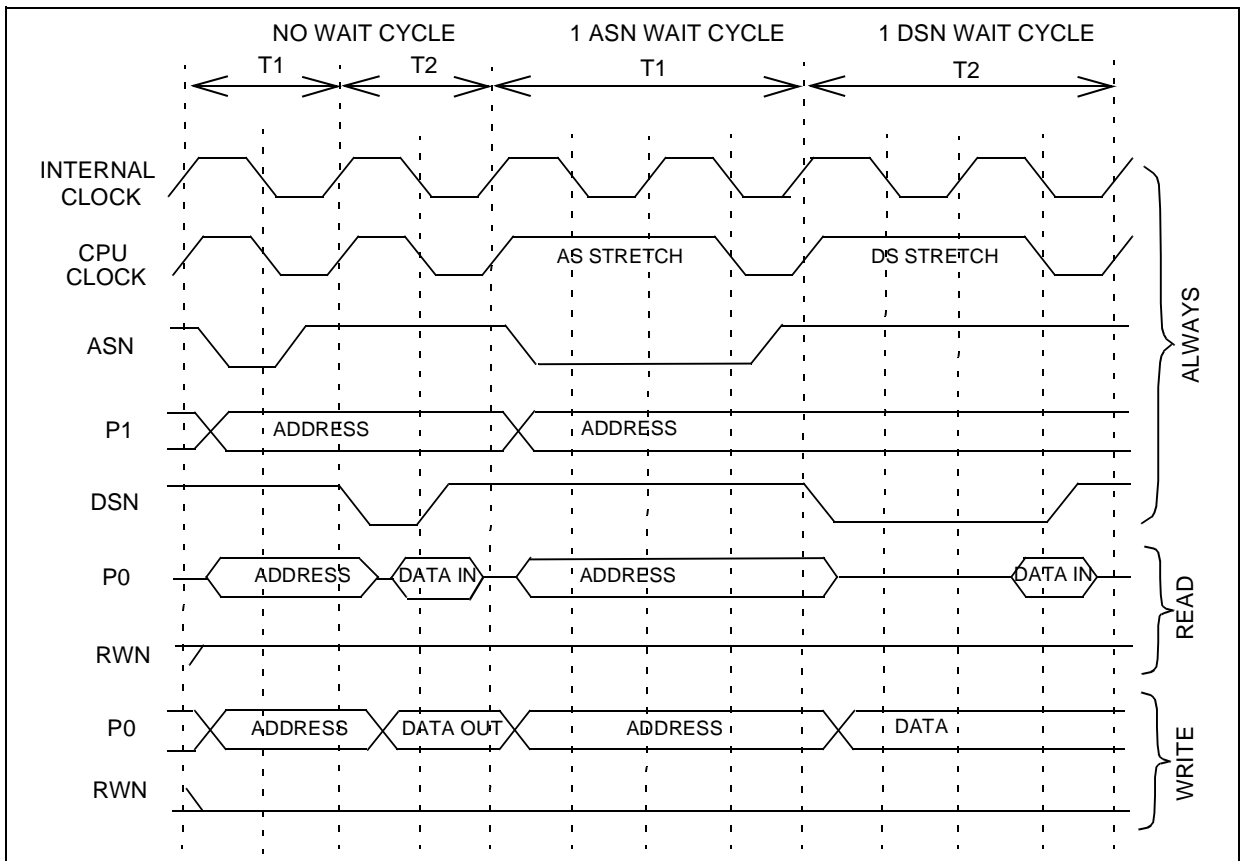
This model is supported by x158, xF120 devices.

In this model, the ST9 is limited to a maximum of 64 Kbytes of directly addressable external memory. It is due to the fact that you can only output 16-bit addresses, which corresponds to a 64 Kbyte external memory space. The signals used in the small memory model interface are the following:

- I/O Port 0 (8 pins): outputs the LSB of the address (A[0..7]) multiplexed with the 8 bits of data (D[0..7]).
- I/O Port 1 (8 pins): outputs the MSB of the address (A[8..15]).
- Address Strobe ASn: is active during clock phase T1. An ASn rising edge indicates that the memory address and control signals are valid.
- Data Strobe DSn: is active during clock phase T2. During an external memory read cycle, the data on Port 0 must be valid before the rising edge of DSn. During an external memory write cycle, the data on Port 0 are output on the falling edge of DSn and they are valid on the rising edge of DSn.
- Read/Write RWn: identifies the type of memory cycle. RWn=1 identifies a Read cycle and RWn=0 identifies a Write cycle.

If you want to access more than 64 Kbytes of memory with this interface, you will have to use some I/O pins as chip select signals for your external memories and handle the management of these pins by software.

Figure 1. External memory Read/Write with and without a programmable wait for MC = 0



## 1.2 LARGE MEMORY MODEL

This model is supported by x250, x150 and x124 devices.

In this model, you can address all the ST9 memory space (4Mbytes), using a 22-bit address bus output on the ST9 ports.

The signals used in the large memory model interface are the following:

- I/O Port 0 (8 pins): same function as small memory model (A[0..7]/D[0..7]).
- I/O Port 1 (8 pins): same function as small memory model (A[8..15]).
- I/O Port 9 (6 pins): outputs the MMU bits of the address (A[16..21]).
- Address Strobe ASn: same function as small memory model.
- Data Strobe DSn: same function as small memory model.
- Data Strobe 2 DS2n: if enabled, this pin can be used as a second Data Strobe. In this case, DSn addresses the external upper memory block while DS2n addresses the external lower memory block. The upper memory is located at addresses >200000h (A21=1, typically RAM), and the lower memory is located at addresses <200000h (A21=0, typically ROM).
- Read/Write RWN: same function as small memory model.

## 2 EXTERNAL MEMORY INTERFACE CONFIGURATION

The external memory interface configuration consists of port configuration, mode configuration, timing configuration, and MMU (Memory Management Unit) configuration. These different types of initialization are presented in the following paragraphs.

Figure 2 shows the mapping of the different registers which are referred to in this chapter.

Figure 2. Register Map

	GROUP E			GROUP F PAGE 21		GROUP F PAGE 2		GROUP F PAGE 3		
				FFh		Reserved		P7DR	R255	
				FEh		P3C2		P7C2	R254	
				FDh		P3C1		P7C1	R253	
				FCh		P3C0		P7C0	R252	
EBh	MODER	R235		FBh		Reserved		P6DR	R251	
				FAh		P2C2		P6C2	R250	
				F9h	DMASR	P2C1		P6C1	R249	
				F8h	ISR	P2C0		P6C0	R248	
				F7h		Reserved		Reserved	R247	
				F6h	EMR2	P1C2		P5C2	R246	
E5h	P5DR	R229	F5h	EMR1		P1C1		P5C1	R245	
E4h	P4DR	R228	F4h	CSR		P1C0		P5C0	R244	
E3h	P3DR	R227	F3h	DPR3		Reserved		Reserved	R243	
E2h	P2DR	R226	F2h	DPR2		P0C2		P4C2	R242	
E1h	P1DR	R225	F1h	DPR1		P0C1		P4C1	R241	
E0h	P0DR	R224	F0h	DPR0		P0C0		P4C0	R240	

Moreover, the WCR register is mapped to R252 (FCh) in register page 0 in Group F. The register map described above is the default setting. If the DPRREM bit in the EMR1 register is set, the mapping of registers P0DR, P1DR, P2DR, and P3DR is exchanged with that of DPR0, DPR1, DPR2, and DPR3.

### 2.1 PORT CONFIGURATION

The different ports used to interface with the external memory must be configured as Alternate Function Output Push-Pull. This concerns Ports 0,1 and Port9

This configuration is done by writing the following values in the three I/O port configuration registers:

- PxC2 = 00h
- PxC1 = FFh
- PxC0 = FFh

(where x is the number of the corresponding I/O port).

After Reset, if your microcontroller is a Romless version, this configuration will be the default one. In this case, the microcontroller must directly address external memory just after the Reset, as there is no internal memory.

For other microcontroller versions (not Romless), the Reset configuration for the I/O ports is usually 'Bidirectional Open-Drain Weak Pull-up' (refer to your device datasheet for special I/O Ports Reset configuration). In this case, you have to initialize the three I/O port configuration registers by software as shown above, before being able to address external memory.

**Note:** By default, the memory control signals (/AS, /DS, /RW) don't need any configuration, because they are not mapped on I/O port pins. So these signals are output as soon as the microcontroller starts accessing external memory.

However, in some microcontrollers (like the ST90158), you have to configure some of these external memory interface signals. For example, to configure the RWn signal for the ST90158 microcontroller, you must select 'Alternate Function Push-Pull' for the P6.5 pin.

### 2.2 MODE CONFIGURATION

The various modes of the external memory interface are configured through the bits of the EMR1 and MODER control registers.

#### ■ EMR1 register:

- Mode Control (MC bit 6): by setting this bit, the Intel mode is used for the external memory interface. In this mode, the ASn pin becomes ALE (Address Load Enable), which corresponds to ASn inverted. The DSn pin becomes OEN (Output Enable), which behaves like DSn during a Read cycle, but which is forced to 1 during Write cycles. The RWn pin becomes WEN (Write Enable), which behaves like DSn during a Write cycle, but which is forced to 1 during Read cycles.
- Buffer Size (BSZ bit 1): when this bit is 0 (default value), the external memory interface pins use smaller, less noisy output buffers. This may limit the operation frequency of the device. Typically, for an internal frequency greater than 10MHz, this bit must be set to use larger (but more noisy) buffers. Another possibility is to add some wait states to slow down the external memory interface signals (see Section 2.4).

**Note:** Some devices don't have the BSZ bit and it is set to 0 by hardware. Please refer to the datasheet.

- Data Strobe 2 Enable (DS2EN bit 5): Large Memory Model option. Setting this bit enables the second Address Strobe pin DS2n. In this case, DSn is used when the upper memory block is addressed (while DS2n is forced to 1), and DS2n is used when the lower memory block is addressed (while DSn is forced to 1).

#### ■ MODER register:

- High Impedance (HIMP bit 0): setting this bit forces the external memory interface signals (Ports 0, 1 and ASn, DSn, RWn) into High Impedance state. This option is recommended for program phases where only internal memory is used, in order to reduce noise. This bit must be kept to zero when the external memory has to be accessed (permanently for a Romless version).

#### ■ EMR2 register **WARNING:**

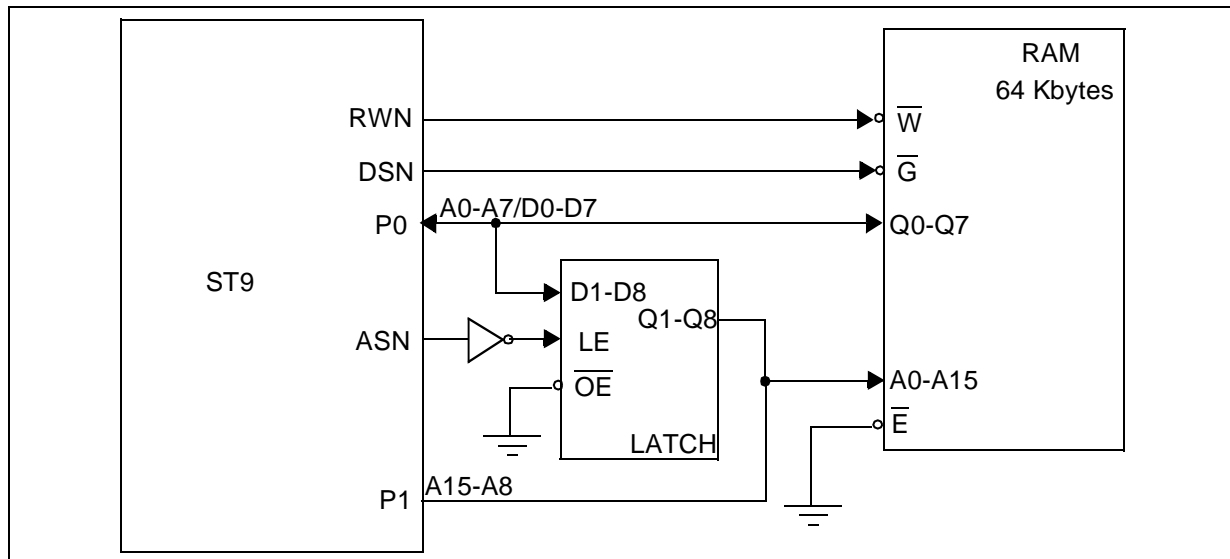
The bit 4 of the EMR2 register must be set by the user when using the external memory interface. Be very careful with this bit because the reset value is 0 in some devices.

2.3 HARDWARE IMPLEMENTATION EXAMPLES

Depending on the mode configuration (and the availability for your device), the hardware implementation can be different.

For a small memory model, you will have to use a latch connected to the multiplexed Address/Data bus. The figure below shows an example of this (Figure 3).

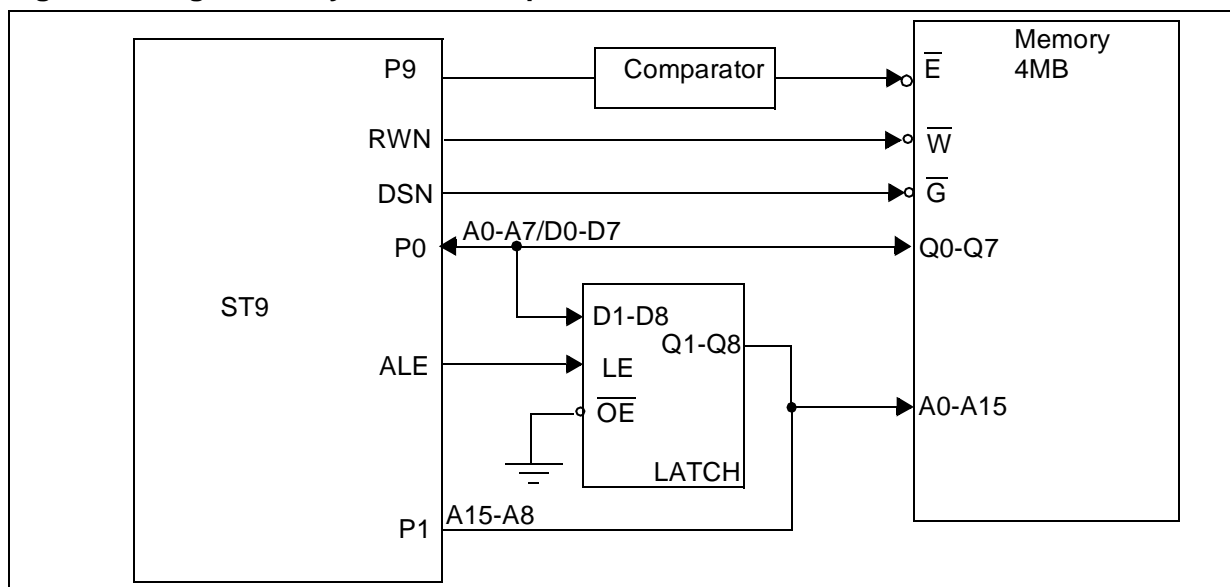
Figure 3. Small Memory Model Example (For MC = 0)



For a ST9 microcontroller with the large External Memory Interface model, you can benefit from the possibilities of the extended mode configuration.

The following figure shows an example of large memory model.

Figure 4. Large Memory Model example for MC = 1



### 2.4 TIMING CONFIGURATION

All the different signals of the external memory interface are driven by the CPU clock (CPUCLK). The CPU clock is the result of the Internal clock (INTCLK) divided by a prescaler. Please refer to the RCCU (Reset and Clock Control Unit) chapter of the ST9 datasheet for further details on the clock.

If your external memory is too slow to follow the ST9 frequency, you can slow down the external memory interface signals in different ways:

- Reducing only the CPUCLK frequency (through the MODER register).
- Adding wait states on the ASn and DSn signals (through EMR2 and WCR registers).
- Adding wait cycles through the WAITn external pin.

#### ■ MODER register:

The three bits PRS0, PRS1, PRS2 load the prescaler division factor for the internal clock (INTCLK which feeds the peripherals). The resulting signal is the CPU clock which drives the external memory interface signals. You can divide the frequency by a factor from 1 to 8, but this slow down will also affect all the code execution, as the CPU clock feeds the core.

#### ■ EMR2 register:

This register contains the control bits used to add wait states on the Address Strobe ASn signal. Two bits (LAS[1:0]) contains the number of clock cycles to add to the CPULCK to stretch ASn during external lower memory block accesses, whereas two other bits (UAS[1:0]) have the same action for external upper memory block accesses. You can add from 0 to 3 wait states to the ASn signal.

Refer to [Figure 1](#) to see the effect of these wait states on the various signals.

#### ■ WCR register:

This register contains the control bits used to add wait states on the Data Strobe DSn signal. Three bits (LDS[2:0]) are used for the lower memory block accesses and three other bits (UDS[2:0]) are used for the upper memory block accesses. You can add from 0 to 7 wait states to the DSn signal.

Refer to [Figure 1](#) to see the effect of these wait states on the various signals.

#### ■ WAITn external pin:

You can also add wait states using the WAITn external pin which indicates to the ST9 that the external memory requires more time to complete the memory access cycle. This function is enabled if the EWEN bit of the EIVR register is set.

This pin is sampled on each rising edge of the internal clock: if WAITn is active (active low), one clock cycle is added to the memory cycle. On the following rising edge of the clock, WAITn is sampled again to continue or finish the memory cycle stretching. If WAITn is sampled active during phase T1 then ASn is stretched, while if WAITn is sampled active during phase T2 then DSn is stretched.



### ■ Tips:

- The wait cycles added on the clock always refer to the internal clock (INTCLK) and not to the CPUCLK. It means that if you choose 3 wait states for example, there will be three INTCLK clock cycles added to the CPU clock (CPUCLK). During the wait states, the CPU-CLK will remain high for the number of periods of INTCLK corresponding to the number of wait states programmed (3 in this example).
- Be careful of the reset values of the control bits in the registers EMR2 and WCR (LAS, UAS, LDS, UDS). By default, the maximum number of wait states is inserted. So, if your memory can work faster, you must remember to change the value of these bits to increase the speed of the ST9 external memory interface.

## 2.5 MMU CONFIGURATION

You will also have to configure your Memory Management Unit to access your external memory. This will be exactly the same as when you configure your internal memory:

- You have to load the correct value in the MMU registers in order to point to your corresponding external memory pages or segments (DPR0, DPR1, DPR2, DPR3, CSR, ISR, DMASR registers).
- You have to add your external memory description and mapping in your scriptfile.
- Don't forget to describe your external memory also in the emulator configuration file 'hardware.gdb'.

Please refer to the ST9 microcontroller datasheet, to the GNU C Compiler User Manual, and to the Emulator User Manual for further details on these configurations.

## 3 TIMING PARAMETERS

Here are the timing parameters and timing diagram for MC =1

### EXTERNAL BUS TIMING TABLE (MC = 1, BSZ =0) (Measured on F250)

( $V_{DD} = 5V \pm 10\%$ ,  $T_A = -40^\circ C$  to  $+125^\circ C$ ,  $C_{load} = 50pF$ )

N°	Symbol	Parameter	Value (Note)			Unit
			Formula	Min.	Max	
1	TsA (ALE)	Address Set-up Time before ALE ↓	$Tck * Wa + TckH - 48$	160		ns
2	ThALE (A)	Address Hold Time after ALE ↓	$TckL - 15$	26		ns
3	TwALE	ALE High Pulse Width	$Tck * Wa + TckH - 58$	150		ns
4	TdAz (OEN)	Address Float (P0) to $\overline{OEN}$ ↓	0	0		ns
5	TdOEN(Az)	P0 driven after $\overline{OEN}$ ↑	$TckL - 13$	29		ns
6	TwOEN	$\overline{OEN}$ Low Pulse Width	$Tck * Wd + TckH - 36$	172		ns
7	TwWEN	$\overline{WEN}$ Low Pulse Width	$Tck * Wd + TckH - 36$	172		ns
8	TdOEN (DR)	$\overline{OEN}$ ↓ to Data Valid Delay	$Tck * Wd + TckH - 44$		164	ns
9	ThDR (OEN)	Data hold time after $\overline{OEN}$ ↑	0	0		ns
10	ThOEN(A)	Address (A21:A8) hold time after $\overline{OEN}$ ↑	0	0		ns
11	ThWEN(A)	Address (A21:A8) hold time after $\overline{WEN}$ ↑	0	0		ns
12	TvA(OEN)	Address (A21:A0) valid to $\overline{OEN}$ ↑	$Tck (Wd + Wa + 1.5) - 76$	382		ns
13	TvA(WEN)	Address (A21:A0) valid to $\overline{WEN}$ ↑	$Tck (Wd + Wa + 1.5) - 44$	414		ns
14	TsD (WEN)	Data Set-up time before $\overline{WEN}$ ↑	$Tck * Wd + TckH - 158$	50		ns
15	ThWEN(DW)	Data Hold Time after $\overline{WEN}$ ↑	$TckL - 15$	26		ns
16	TdALE (WEN)	ALE ↑ to $\overline{WEN}$ ↑ Delay	$Tck (Wd + Wa + 1.5) - 54$	404		ns
17	TdALE (OEN)	ALE ↑ to $\overline{OEN}$ ↑ Delay	$Tck (Wd + Wa + 1.5) - 50$	408		ns

**Notes:** The expressions in the Formula column show how to calculate the typical parameter value depending on the CPU Clock period and the number of inserted wait cycles.  
The values in the Measured column give the parameter values for a CPU clock at 12 MHz and two wait states for T1 and T2.

#### Legend:

Tck = INTCLK period = OSCIN period when OSCIN is not divided by 2;  
2\*OSCIN period when OSCIN is divided by 2;  
OSCIN period / PLL factor when the PLL is enabled

TckH = INTCLK high pulse width (normally = Tck/2, except when INTCLK = OSCIN, in which case it is OSCIN high pulse width)

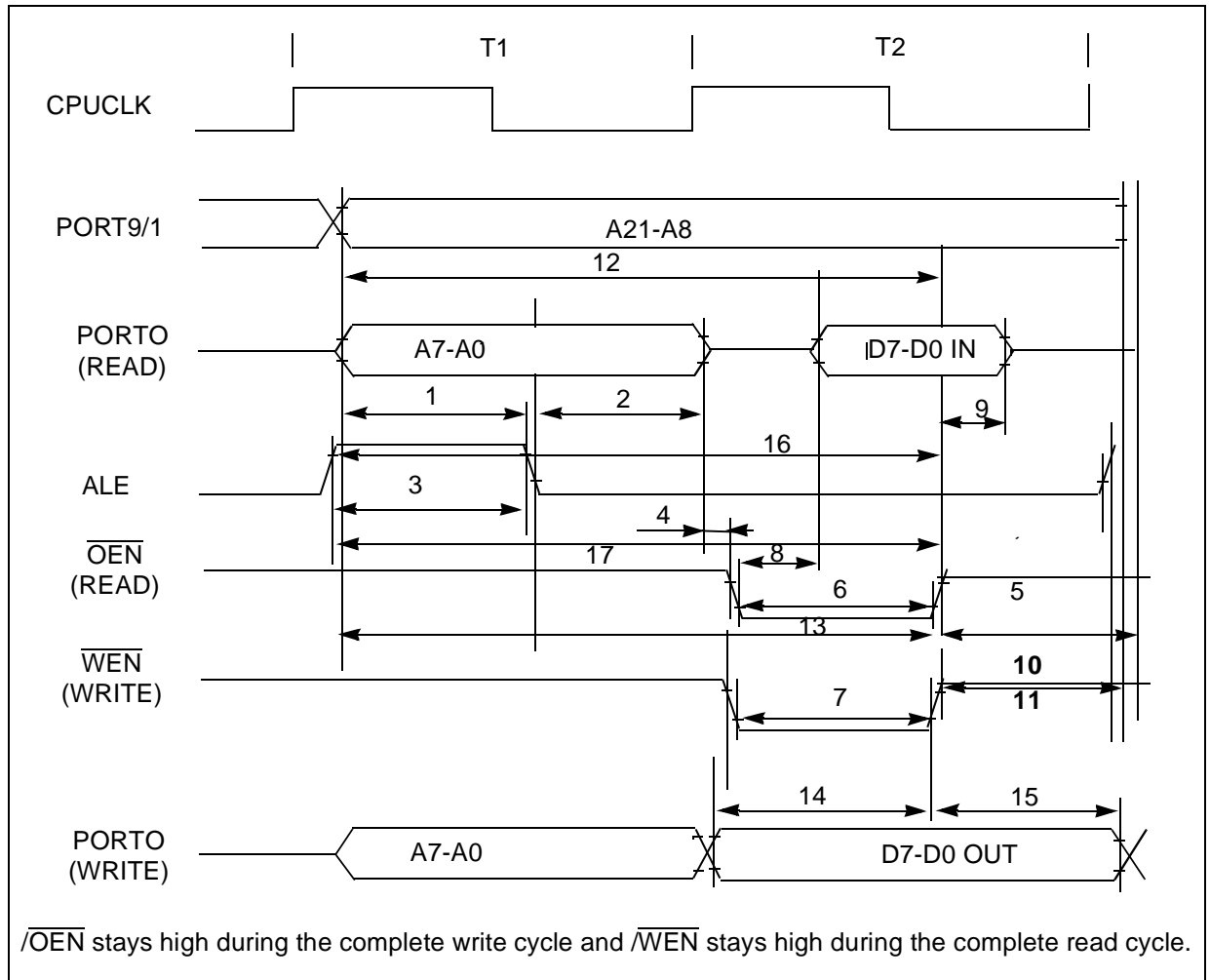
TckL = INTCLK low pulse width (normally = Tck/2, except when INTCLK = OSCIN, in which case it is OSCIN low pulse width)

P = clock prescaling value (=PRS; division factor = 1+P)

Wa = wait cycles on ALE; = max (P, programmed wait cycles in EMR2, requested wait cycles with WAIT)

Wd = wait cycles on  $\overline{OEN}$  and  $\overline{WEN}$ ; = max (P, programmed wait cycles in WCR, requested wait cycles with WAIT)

Figure 5. Timing Diagram for MC =1



**Note:** Refer to [Figure 1](#) for the timing diagram for MC=0

## EXTERNAL BUS TIMING TABLE (MC = 1, BSZ = 1)(Measured on F250)

(V<sub>DD</sub> = 5V ± 10%, T<sub>A</sub> = -40°C to +125°C, Cload = 50pF)

N°	Symbol	Parameter	Value (Note)			Unit
			Formula	Min.		
1	TsA (ALE)	Address Set-up Time before ALE ↓	Tck*Wa+TckH - 28	180		ns
2	ThALE (A)	Address Hold Time after ALE ↓	TckL - 15	26		ns
3	TwALE	ALE High Pulse Width	Tck*Wa+TckH - 24	184		ns
4	TdAz (OEN)	Address Float (P0) to $\overline{\text{OEN}}$ ↓	0	0		ns
5	TdOEN(Az)	P0 driven after $\overline{\text{OEN}}$ ↑	TckL - 22	20		ns
6	TwOEN	$\overline{\text{OEN}}$ Low Pulse Width	Tck*Wd+TckH - 24	184		ns
7	TwWEN	$\overline{\text{WEN}}$ Low Pulse Width	Tck*Wd+TckH - 16	192		ns
8	TdOEN (DR)	$\overline{\text{OEN}}$ ↓ to Data Valid Delay	Tck*Wd+TckH - 36		172	ns
9	ThDR (OEN)	Data hold time after $\overline{\text{OEN}}$ ↑	0	0		ns
10	ThOEN(A)	Address (A21:A8) hold time after $\overline{\text{OEN}}$ ↑	0	0		ns
11	ThWEN(A)	Address (A21:A8) hold time after $\overline{\text{WEN}}$ ↑	0	0		ns
12	TvA(OEN)	Address (A21:A0) valid to $\overline{\text{OEN}}$ ↑	Tck (Wd+Wa+1.5) - 48	410		ns
13	TvA(WEN)	Address (A21:A0) valid to $\overline{\text{WEN}}$ ↑	Tck (Wd+Wa+1.5) - 42	416		ns
14	TsD (WEN)	Data Set-up time before $\overline{\text{WEN}}$ ↑	Tck*Wd+TckH - 68	140		ns
15	ThWEN(DW)	Data Hold Time after $\overline{\text{WEN}}$ ↑	TckL - 15	26		ns
16	TdALE (WEN)	ALE ↑ to $\overline{\text{WEN}}$ ↑ Delay	Tck (Wd+Wa+1.5) - 34	424		ns
17	TdALE (OEN)	ALE ↑ to $\overline{\text{OEN}}$ ↑ Delay	Tck (Wd+Wa+1.5) - 42	416		ns

**Note:** The timing diagram remains the same for BSZ = 1. We don't have these measurements for MC= 0 because we want to promote MC = 1 mode in order to conform with industry standard.

### 3.1 MAKING YOUR OWN MEASUREMENTS

Most of these parameters are easy to measure just by writing small programs for writing or reading to/from the memory. Except for the Data valid to /OEN falling parameter which requires capturing the instant data seen by the microcontroller during a read cycle. To measure this, a pulse generator is required which can output a pulse with a configurable delay from a given signal (ALE in this case.) This pulse is used to trigger one of the P0 port pins and data on port0 is dumped to port4 continuously. So by monitoring port4 and adjusting the delay to get a '1' at port4, keeping the pulse as narrow as possible the actual moment at which data becomes valid for the microcontroller with respect to ALE going low can be captured.

All the measurements for MC=1 have been made between 80% transition of first edge and 20% transition of the second so as to get the minimum values for the measured parameter. By looking at the parameter values for BSZ = 0. It can be concluded that its necessary to add 1 or 2 wait states while operating at high frequencies such as 12MHz otherwise the value of some of the parameters becomes negative.

## 4 EXTERNAL MEMORY INTERFACE FOR OLD ST9 DEVICES

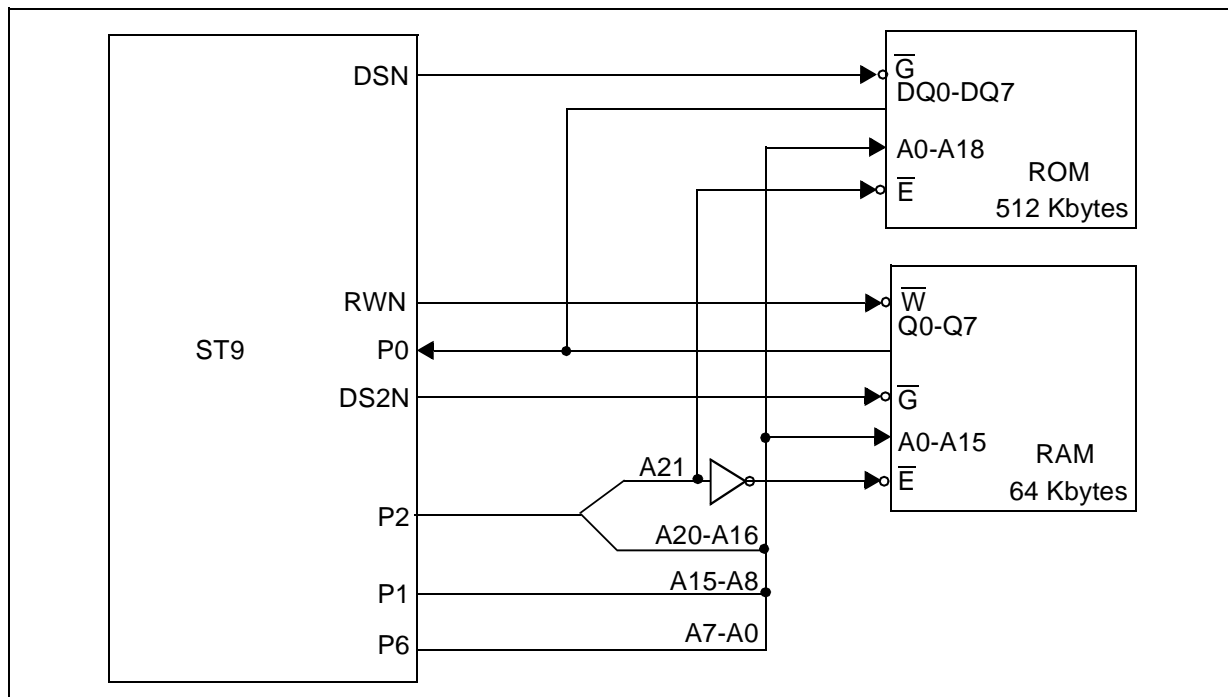
- Some older ST9 devices output the MSB of the address on port2 and LSB on either port0/ port 6 based on the NMB bit:
- I/O Port 2 (6 pins): outputs the MMU bits of the address (A[16..21]).
- I/O Port 6 (8 pins): if the non-multiplexed bus option is chosen, this port outputs the LSB of the address (A[0..7]). In this case, Port 0 only outputs the 8 bits of data (D[0..7]).

### 4.1 NON-MULTIPLEXED BUS (NMB BIT 3): LARGE MEMORY MODEL OPTION

When this bit is zero, Port 0 outputs the multiplexed data and the LSB of the address. When this bit is 1, it is port 6 which outputs the address LSB (Port 0 is released in high impedance during this time), and Port 0 only outputs data.

Figure 6 shows an example for a Large Memory Model, where the non-multiplexed bus (NMB bit of the EMR1 register) and the second Data Strobe (DS2EN bit of the EMR1 register) options have been chosen. This allows some external hardware devices like the latch to be saved.

Figure 6. Hardware Implementation: Large Memory Model Example



### 5 SOFTWARE EXAMPLE

The code below shows a software example of a Large Memory Model interface (x150, x250 microcontrollers) with an external memory located in segment 24 of memory (and pointed to by DPR3).

```
spp #21
ld R245, #082h ;EMR1 register: Normal mode & high-speed buffers
ld R246, #050h; ;EMR2 register: zero wait states & (Bit 4)=1

spp #0
ld R252, #40h ; WCR: zero wait states
ld R235, #20h ; MODER: No prescaler division & No High Impedance

spp #3
ld R250, #000h ;P6.5 in Alternate function
ld R249, #0FFh ; Push-Pull
ld R248, #0FFh ; (RW pin)
spp #2
ld R242, #000h ;Port 0 in Alternate function
ld R241, #0FFh ; Push-Pull
ld R240, #0FFh ; (Address LSB/Data multiplexed)
ld R246, #000h ;Port 1 in Alternate function
ld R245, #0FFh ; Push-Pull
ld R244, #0FFh ; (Address MSB)

spp #43
ld R254, #000h ;Port 9 in Alternate function
ld R253, #0FFh ; Push-Pull
ld R252, #0FFh

spp #21
ld R240, #0x00 ;DPR0 register (Segment 0 for internal flash)
ld R241, #0x01 ;DPR1 register (Segment 0 for internal flash)
ld R242, #0x80 ;DPR2 register (Segment 20 for Internal RAM
ld R243, #0x96 ;DPR3 register (Segment 24 for External RAM)

sdm ; set data memory
ld 0xC000, #0xAA ;examples of data transfers
ld 0xC001, #0xAA ; in external memory
ld 0xC002, #0xAA ;
```

“THE PRESENT NOTE WHICH IS FOR GUIDANCE ONLY AIMS AT PROVIDING CUSTOMERS WITH INFORMATION REGARDING THEIR PRODUCTS IN ORDER FOR THEM TO SAVE TIME. AS A RESULT, STMICROELECTRONICS SHALL NOT BE HELD LIABLE FOR ANY DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES WITH RESPECT TO ANY CLAIMS ARISING FROM THE CONTENT OF SUCH A NOTE AND/OR THE USE MADE BY CUSTOMERS OF THE INFORMATION CONTAINED HEREIN IN CONNECTION WITH THEIR PRODUCTS.”

Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics.

All other names are the property of their respective owners

© 2004 STMicroelectronics - All rights reserved

STMicroelectronics GROUP OF COMPANIES

Australia – Belgium - Brazil - Canada - China – Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan -  
Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States

[www.st.com](http://www.st.com)