Application Note, V 1.1, September 2002

AP32035

TC1775

External memory interface

32-Bit Single-Chip Microcontroller

Microcontrollers

Infineon technologies

Never stop thinking.

Revision History:	**2002-09**	V 1.1
Previous Version:	2001-09	V 1.0

Page	Subjects (major changes since last revision)
all	Changed to Infineon Template

We Listen to Your Comments

Any information within this document that you feel is wrong, unclear or missing at all? Your feedback will help us to continuously improve the quality of this document. Please send your proposal (including a reference to this document) to:

mcdocu.comments@infineon.com

Edition 2002-09

Published by
Infineon Technologies AG
81726 München, Germany

LEGAL DISCLAIMER

THE INFORMATION GIVEN IN THIS APPLICATION NOTE IS GIVEN AS A HINT FOR THE IMPLEMENTATION OF THE INFINEON TECHNOLOGIES COMPONENT ONLY AND SHALL NOT BE REGARDED AS ANY DESCRIPTION OR WARRANTY OF A CERTAIN FUNCTIONALITY, CONDITION OR QUALITY OF THE INFINEON TECHNOLOGIES COMPONENT. THE RECIPIENT OF THIS APPLICATION NOTE MUST VERIFY ANY FUNCTION DESCRIBED HEREIN IN THE REAL APPLICATION. INFINEON TECHNOLOGIES HEREBY DISCLAIMS ANY AND ALL WARRANTIES AND LIABILITIES OF ANY KIND (INCLUDING WITHOUT LIMITATION WARRANTIES OF NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS OF ANY THIRD PARTY) WITH RESPECT TO ANY AND ALL INFORMATION GIVEN IN THIS APPLICATION NOTE.

Information

For further information on technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies Office (**www.infineon.com**).

Warnings

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies Office.

Infineon Technologies Components may only be used in life-support devices or systems with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system, or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body, or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

Table of Contents Page

Table of Contents **Page**

1 Conventions, Definitions and Abbreviations

Name	Description
Byte	8-bit data format quantity
Half-word	16-bit data format quantity
Word	32-bit data format quantity
kByte	1024 bytes of memory
MByte	1048576 bytes of memory
BCU	Bus Control Unit
CPU	Central Processing Unit
DMU	Data Memory Unit
EBU	External Bus Unit
PCP	Peripheral Control Processor
PMU	Program Memory Unit
SCU	System Control Unit

2 Introduction

The External Bus Unit (EBU) of the TC1775 is the interface to external memories and peripheral units which use the external address and data bus. The EBU is primarily used for communication with external memories or peripheral units via the FPI Bus and also for instruction fetches directly from the PMU if external Burst Flash memories will be used. The EBU controls all transactions required for these operations. In Burst Mode the instruction fetches will be transferred directly from the External Bus Unit (EBU) to the Program Management Unit (PMU) without using the FPI bus. During these direct instruction fetches, the FPI Bus can be used for transfers of peripheral units like Peripheral Control Processor, Analog-Digital Converter, General Purpose I/O's, Communication interfaces etc.

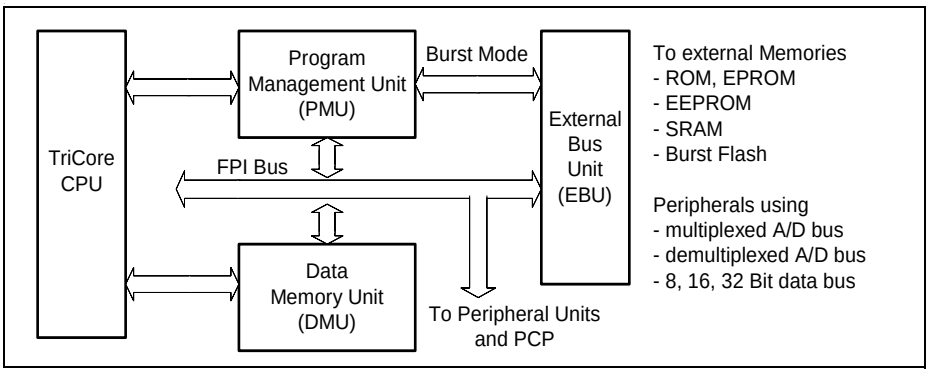

Figure 1 TC1775 External Bus Unit

This document describes how to connect different types of memories to the External Bus Unit interface. The most important settings of the configuration registers and examples how to initialize external memories can also be found within this document. Additional information and a detailed description of all registers can be found in the System Units part of the TC1775 User's manual.

3 External Bus Unit of the TriCore TC1775

3.1 I/O voltage ranges

The EBU is connected to Port 4..0 and builds a communication interface to external memories and peripheral units. The operation voltage for the bus output buffers is 2.5 Volt connected to pin V_{DDP05}. Input receivers work also with an internal supply voltage of 2.5 Volt but are 3.3 Volt tolerant (Class B).

After RESET the EBU is **enabled** by default and can be enabled or disabled by Bit DISS (Bit 1) in the EBU Clock control register EBU_CLC.

Port	Signals	Purpose	Operation
Port 0	D[15:0] or	Lower Data bus in demultiplexed mode	2.5 V
	AD[15:0]	Lower Address/Data bus in multiplexed mode	
Port 1	D[31:16] or	Higher Data bus in demultiplexed mode	2.5 V
	AD[31:16]	Higher Address/Data bus in multiplexed mode	
Port 2	A[15:0]	Lower Address bus in demultiplexed mode	2.5 V
Port 3	A[25:16]	Higher Address bus in demultiplexed mode	2.5 V
	$\overline{\text{CS}}$[3:0],	Chip Select signals (active low)	
	$\overline{\text{CSEMU}}$, $\overline{\text{CSOVL}}$	Emulation and Overlay support (active low)	
Port 4	$\overline{\text{RD}}$, RD/$\overline{\text{WR}}$	Read control, Write control	2.5 V
	ALE	Address latch enable	
	$\overline{\text{ADV}}$	Address valid output	
	$\overline{\text{BAA}}$	Burst address advance output	
	$\overline{\text{BC}}$[3:0]	Byte control line 3..0	
	$\overline{\text{WAIT}}$/$\overline{\text{IND}}$	Wait input, End of burst input	

Table 1 EBU Port overview and description

The TC1775 uses separate power supply pins for Core supply and Port supply. The Port Power supply voltage has to be connected to pin V_{DDP05} = 2.5 Volt for Port 5..0 and to pin V_{DDP813} = 3.3 Volt - 5 Volt for Ports 13..8. The pins for address and data bus and also for control signals are specified for V_{DDP05} = 2.3 .. 2.75 Volt and all input pins are 3.3 Volt tolerant. All external memories which use a 3.3 Volt power supply for the V_{DDQ} output voltage or a single power supply V_{DD} for core and I/O power supply can be connected directly to the EBU interface. A Worst-case check of the driver and receiver conditions

are necessary to guarantee a successful system behavior over the complete parameter range.

Parameter	Symbol	Limit value min.	Limit value max.	Conditions
Specification: V_{DDP05} = 2.3 .. 2.75 V				
Output low voltage	V_{OL}		0.45 V	I_{OL} = 600 µA
Output high voltage	V_{OH}	0.9 x V_{DDP05}		I_{OL} = - 600 µA
Input low voltage	V_{IL}	-0.5	0.2 x V_{DDP05}	
Input high voltage	V_{IH}	0.7 x V_{DDP05}	3.7 V	
Default: V_{DDP05} = 2.5 V				
Output low voltage	V_{OL}		0.45 V	I_{OL} = 600 µA
Output high voltage	V_{OH}	2.25 V		I_{OL} = - 600 µA
Input low voltage	V_{IL}	-0.5	0.5 V	
Input high voltage	V_{IH}	1.75 V	3.7 V	

Table 2 Input / Output DC-Characteristics of EBU interface

Depending on the Input/Output DC-characteristics of external memory it could be necessary to increase the supply voltage of the EBU I/O's and to decrease the supply voltage of memory I/O's within the allowed range. Both values must be within the specified values.

Examples:

An external SRAM device is specified for supply voltage V_{CC}=3.3 Volt, V_{IHmin} = 2.0 Volt and V_{ILmax} = 0.8 Volt. Because of the specified DC characteristics of Microcontroller and SRAM, both values are within the specification if the I/O supply voltage of the TC1775 is V_{DDP05} = 2.5 Volt.

An external Flash device is specified with V_{CC}=3.3 Volt, V_{IHmin} = 2.31 Volt and V_{ILmax} = 0.8 Volt. Because of the specified DC characteristics of Microcontroller and Flash the value of V_{OH} is outside the specification if the I/O supply voltage of the TC1775 is V_{DDP05} = 2.5 Volt. In this case an increase of the TC1775 I/O supply voltage V_{DDP05} to 2.6 Volt (V_{OHmin} = 2.34 Volt) and a decrease of the Flash power supply V_{CC} to 3.0 Volt (V_{IHmin} = 0.7 x V_{CC} = 2.1 Volt) meets the specification of both devices.

3.2 EBU External Address Ranges

The EBU reacts to addresses in a range defined as external memory. Each address is compared against the address defined in the Address Select Register **EBU_ADDSELx**.

Segment	Address Range	Description
10	A000 0000$_H$ - AFFF FFFF$_H$	External memory space, cached area, 256 MByte
11	B000 0000$_H$ - BDFF FFFF$_H$	External memory space, non-cached area, 224 MByte
11	BE00 0000$_H$ - BFFF FFFF$_H$	External Emulator space, non-cached area, 16 MByte
14	E000 0000$_H$ - EFFF FFFF$_H$	External Peripheral and data memory space, non-cached area, 256 MByte

Table 3 EBU External address ranges

The EBU provides five programmable address regions. Each region can be controlled by two separated registers, the Address Select register **ADDSELx** and the Bus Control register **BUSCONx**. Each address region is linked to one chip select line $\overline{CSx}$ which will be activated if an address fits to an address region. In the Address Select register the Base Address within the memory map is specified. Configuration and timing parameters defined for the external memory region can be found in the Bus Control register. The smallest possible address region is **4 kBytes** (MASK=15), the largest region can be set to **128 MByte** (MASK=0).

Address Region	Address Select Register	Bus Control Register	Chip Select
User region 0	EBU_ADDSEL0	EBU_BUSCON0	$\overline{CS0}$
User region 1	EBU_ADDSEL1	EBU_BUSCON1	$\overline{CS1}$
User region 2	EBU_ADDSEL2	EBU_BUSCON2	$\overline{CS2}$
User region 3	EBU_ADDSEL3	EBU_BUSCON3	$\overline{CS3}$

Table 4 EBU User address regions

Note: The TC1775 provides an additional Chip select for the emulator region $\overline{CSEMU}$ and an additional Chip select for the overlay memory $\overline{CSOVL}$. $\overline{CSEMU}$ should not be used in any application and not for normal access.

If the External Instruction Fetch Path Select Bit **EXTIF** in Register **SCU_CON** is set to Instruction fetch via FPI Bus (default), the range of address bits which will be compared against the FPI Bus address can be defined in the **MASK** bits of register **ADDSEL**. The base address of the memory range has to be defined in the **BASE** option.

EBU_ ADDSELx	Description	Value or Range	Default value
REGEN	Memory region enable/disable	0_B or 1_B	boot mode
MIRRORE	Memory region mirror enable/disable	0_B or 1_B	0
BASE	Base address of external memory range	A[31:12]	boot mode
MASK	Range and number of address bits compared to BASE[26:12]	0 .. 15	0

Table 5 Definition of Address Select Register ADDSEL

Note: Memory Region Enable Bit and Base Address default value depend on boot mode

EBU_ BUSCONx	Description	Value or Range	Default value
PORTW	Memory data width = 8, 16 or 32 Bit	00_B .. **10_B**	32 Bit
AGEN	multiplexed or demultiplexed mode	**00_B** or 11_B	demuxed
WAITRDC	Number of Wait-States in read access	0 .. 127	48
WAITWRC	Number of Wait-States in write access	0 .. **7**	7
CMULT	Wait cycle multiplier	1, 4, 8, **16**	16
CMULTR	Read cycle multiplier	**1**, 4, 8, 16	1
WAIT	Wait-State insertion = asynchronous, synchronous, variable or disabled	**00_B** .. 11_B	variable
SETUP	Cycle 0 generation	**0_B** or 1_B	no Cycle0
RECOVC	Number of recovery cycles	0..**3**	3
HOLDC	Number of hold cycles in demultiplexed mode	0..**3**	3
WAITINV	$\overline{\text{WAIT}}$ = high or low active	**0_B** or 1_B	active low
BCGEN	Functionality of Byte Control BC[3:0]	00_B .. 10_B	Control
ALEC	Address Latch Enable Duration control	0..**3**	3
WRDIS	Memory Region Write Protection	0_B or **1_B**	disabled

Table 6 Parameter in the Bus Configuration Register BUSCONx

3.2.1 Settings for Address Select and Bus Configuration Registers

3.2.1.1 Default settings

Number of WRITE Wait states: nWS_W = CMULT * WAITWRC = 16 * 7 = 112 cycles
Number of READ Wait states: nWS_R = CMULTR * WAITRDC = 1 * 48 = 48 cycles
Number of HOLD cycles: nHold = CMULT * HOLDC = 16 * 3 = 48 cycles
Number of RECOVERY cycles: nRECOV = CMULT * RECOVC = 16 * 3 = 48 cycles
Number of SETUP cycles: nSETUP = SETUP = 0 cycles

3.2.1.2 Examples

Note: The largest possible address range is limited to 128 MByte.

Description of Configuration	Settings
• Memory range = $A0000000_H$ .. $A03FFFFF_H$ = 4 MByte • use User Address Range 0 connected to $\overline{CS0}$) • A[26:22] will be used for the address range selection • Enable region, Region is not mirrored • Write access enabled • No Hold cycle, no RECOVERY cycle • No READ wait state, no WRITE wait state • 32-Bit demultiplexed address/data bus	`EBU_ADDSEL0=0xA0000051` `EBU_BUSCON0=0x00020000`
• Memory range = $A4000000_H$ .. $A4FFFFFF_H$ = 16 MByte • use User Address Range 0 connected to $\overline{CS1}$) • A[26:24] will be used for the address range selection • Enable region, Region is not mirrored • Write access enabled • No Hold cycle, no RECOVERY cycle • 2 READ wait state, 2 WRITE wait state • 32-Bit demultiplexed address/data bus	`EBU_ADDSEL1=0xA4000031` `EBU_BUSCON1=0x00020180`

Table 7 Settings for Address Select and Bus Configuration Registers

Note: During address range selection the address bits A[31:27] must always match.

4 Bus topologies of external memory devices

The EBU configuration is controlled by EBU Controls / Status registers, Address Region registers and Emulator registers. External memory space can be managed via the four address region registers **EBU_ADDSEL[3:0]**. The corresponding settings for each memory region can be controlled by the control / status register **EBU_BUSCON[3:0]**. The external device data width for the different memory ranges can be adjusted in each EBU bus configuration register EBU_BUSCON. The Settings can be done in the PORTW Field of EBU_BUSCONx[17:16].

Field	Bits	Type	Description
PORTW	[17:16]	rw	**External Device Data Width Control** 00_B 8-Bit data 01_B 16-Bit data 10_B 32-Bit data (default after reset) 11_B reserved

Table 8 PORTW field of EBU Register EBU_BUSCONx

Via the Byte Control signals $\overline{BC}$[3:0] byte access to corresponding byte locations can be controlled so that external memory devices with a data width of 8, 16 or 32 bits can be connected. The number of bus accesses for an instruction fetch from external memory depends from the bus width, so a 32-Bit instruction fetch is divided into four 8-Bit access for a 8-Bit data bus configuration.

For **Burst Flash memory** accesses **only** the **32-Bit** data bus width is supported. In Burst Mode, all instruction fetches will be transferred directly from the External Bus Unit (EBU) to the Program Management Unit (PMU) without using the FPI bus. During these direct instruction fetches, the FPI Bus can be used for transfers of peripheral units like Peripheral Control Processor, Analog-Digital Converter, General Purpose I/O's, Communication interfaces etc..

4.1 Demultiplexed 8-Bit interface

External memories with a data bus width of 8-Bit can be connected directly to the EBU data interface. The data transfer width is 8-Bit for each memory access and the address range of the external memory can be mapped directly to the EBU address interface. The setting for the external bus width of 8-Bit for PORTW is 00_B.

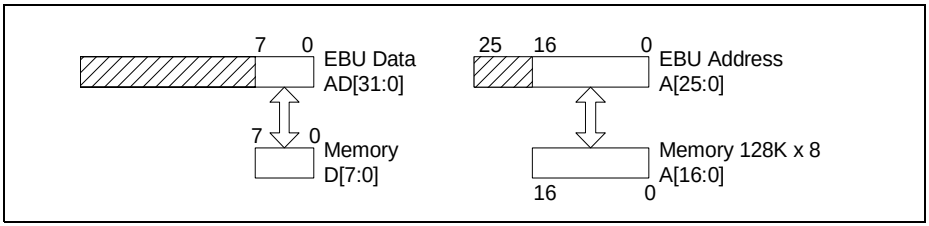

Figure 2 Demultiplexed 8-Bit interface

4.2 Demultiplexed 16-Bit interface

Memory devices in the organization x16 like a SRAM 256Kx16 transfer 16-Bit data for each read access. SRAMs use very often control input pins to enable the output of the upper byte, lower byte or both. A SRAM device expects the configuration on $\overline{HB}$ and $\overline{LB}$ pins. The management of these two pins can be done directly by using the Byte Control Pins of the EBU bus control signals $\overline{BC}$[1:0]. All memory accesses to an external memory device are 16-bit aligned because with each access 16-bit data are transferred on the EBU data bus AD[15:0].

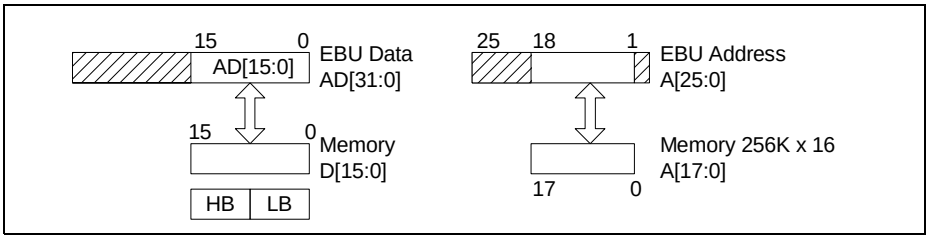

Figure 3 Demultiplexed 16-Bit interface

Because of the data width of 16-bit the address mapping for the EBU address line has to be shifted by one bit so the address lines A[17:0] of an external 256Kx16 SRAM has to be connected to EBU address pins A[18:1]. The setting for the external bus width of 16-Bit for PORTW is 01_B.

4.3 Demultiplexed 32-Bit interface

32-Bit data transfers can be performed by using two devices with a 16-Bit data interface in parallel. Each device transfers half of the maximum data bus width of 32-Bit for each read access. All memory accesses to an external memory device are 32-bit aligned because with each access 32-bit data are transferred on the EBU data bus AD[31:0]. To handle the two external memory devices in parallel the address lines A[17:0] have to be connected to the EBU address lines A[19:2] to create an address offset and map the address lines in the correct way. The management of the upper and lower data bus of each memory device can be done directly by using the Byte Control Pins of the EBU bus control signals /BC[3:0].

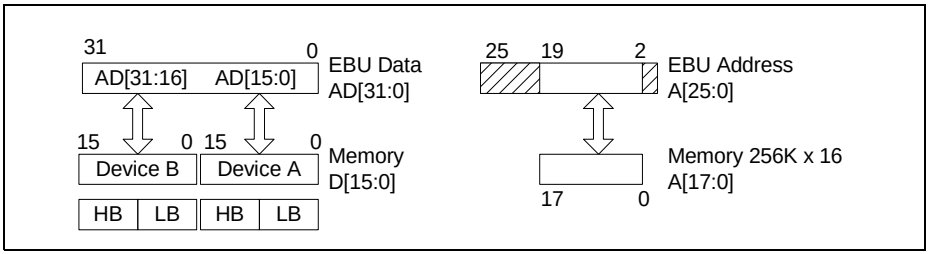

Figure 4 Demultiplexed 32-Bit interface

The setting for the external bus width of 32-Bit for PORTW is 10_B.

Note: In demultiplexed mode an address is driven only on address lines A[25:0].

4.4 Read Access timing for demultiplexed mode

Parameters within the configuration registers of the External Bus Unit (EBU) allow to design applications with different read and write timings for the external memory. The timing of the external memory can be defined by different fields like Wait Cycle Multiplier, Read Cycle multiplier, Read Wait-States, Write Wait-States and the definition of an additional Recovery Cycle, Hold/Pause Cycle (Write) or Extended Address Cycle. These parameters can be found in the EBU Bus Configuration Register **EBU_BUSCONx**.

Cycle	Description	Cycles min	Cycles max	Parameter
Cycle 0	Address setup (optional)	0	1	SETUP
Cycle 1a	Read activation	1	1	fixed
Cycle 1b	Read activation (wait state)	0	127 x 8	WAITRDC[6:0] CMULTR[1:0]
Cycle 2	Read deactivation	1	1	fixed
Cycle 3	Recovery	0	3	RECOVC[1:0]

Table 9 Read Access timing for demultiplexed mode

The minimum read time for a asynchronous read in demultiplexed mode is two clock cycles. Fixed cycles are marked “grey” in the following timing diagrams.

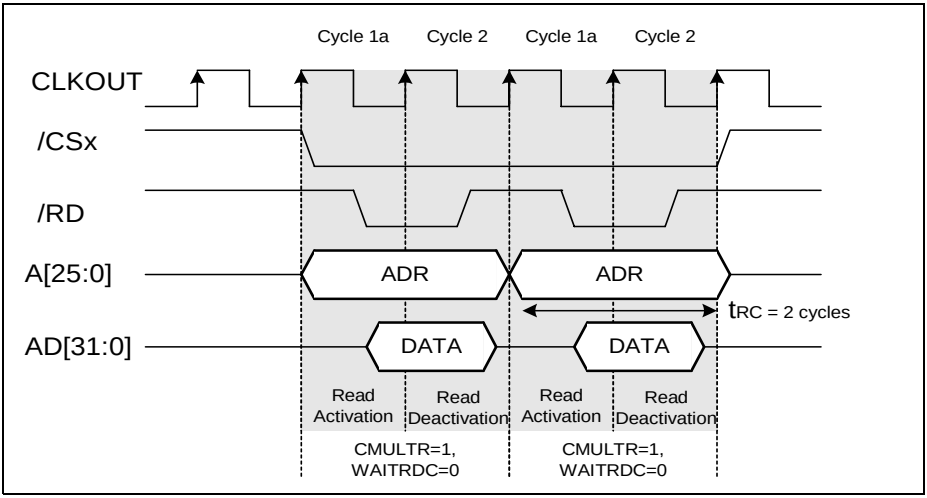

Figure 5 Read access timing in asynchronous, demultiplexed mode (t_{RC}=2)

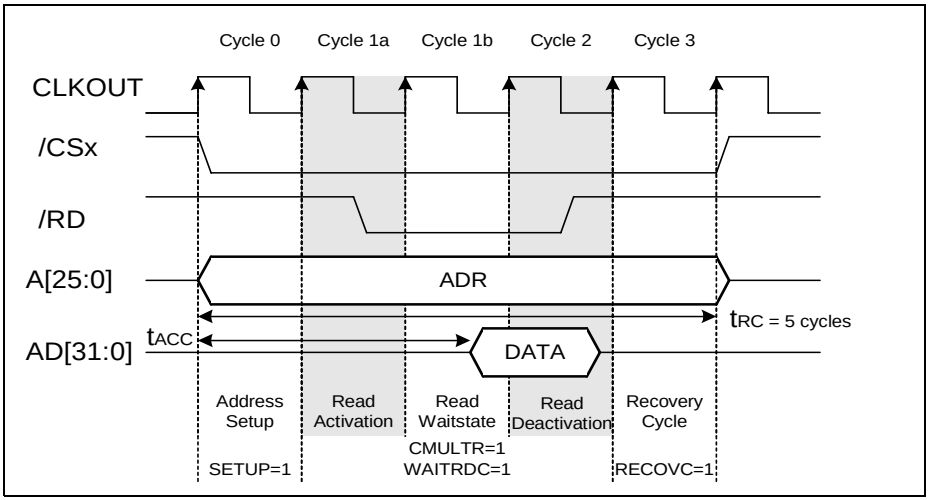

Figure 6 Read access timing in asynchronous, demultiplexed mode (t_{RC}=5)

4.5 Multiplexed Address/Data-Bus

In multiplexed mode both address and data are driven on the multiplexed address/data bus AD[31:0]. In the first part of an access, the address is driven on AD[31:0] together with the Address Latch Enable signal ALE = high.

In the second part in combination with ALE = low, the data is driven by the EBU interface for a write access during RD/$\overline{WR}$ = low and $\overline{RD}$ = high. For a read access, data will be driven during $\overline{RD}$ = low and RD/$\overline{WR}$ = high. Wait cycles can be inserted between address and data cycle by initializing the WAITRDC, CMULTR, WAITWRC and CMULT fields in the selected EBU_BUSCONx register.

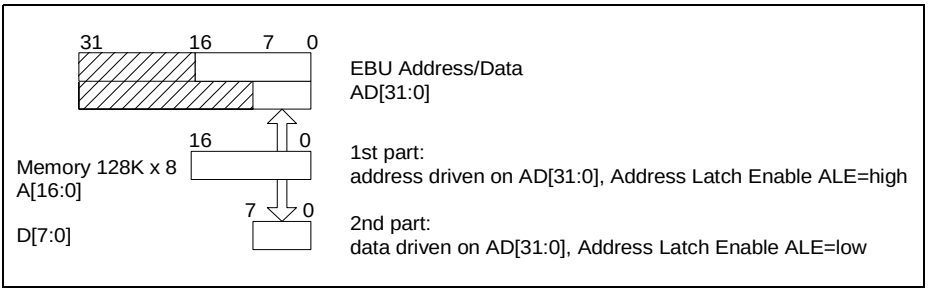

Figure 7 Multiplexed Address/Data-Bus

4.6 Read Access timing for multiplexed mode

Cycle	Description	Cycles min	Cycles max	Parameter
Cycle 0a	Address setup	1	1	fixed
Cycle 0a	Address setup (repeat)	0	3	SETUP
Cycle 1	Address hold	1	1	fixed
Cycle 2a	Read activation	1	1	fixed
Cycle 2b	Read activation (wait state)	0	127 x 8	WAITRDC[6:0] CMULTR[1:0]
Cycle 3	Read deactivation	1	1	fixed
Cycle 4	Recovery	0	3	RECOVC[1:0]

Table 10 Read Access timing for multiplexed mode

The minimum read time for an asynchronous read in multiplexed mode is four clock cycles.

Note: In demultiplexed mode, only addresses A[25:0] are driven to the external bus. In multiplexed mode, a complete 32-Bit address is driven to AD[31:0].

5 EBU configuration

5.1 Booting from external memory

The EBU supports booting from external memory using its default settings for setup and timing (see **Chapter 3.2 'EBU External Address Ranges"** and **Chapter 3.2.1 'Settings for Address Select and Bus Configuration Registers"**). With these settings a external boot memory configuration word can be read from address <BOOTBASE + 0x04>. The settings of the EBU configuration register EBU_BUSCON0 will be overwritten with values read from external memory and set the EBU configuration to proper values for additional reads from external memory. The BOOTCFG register is located in the external boot memory. The options for a external boot sequence are asynchronous demultiplexed mode, memory connected to $\overline{CS0}$. The EBU use registers EBU_ADDSEL0 and EBU_BUSCON0 and operates as a external bus master.

Option	Field in BOOTCFG	Field in EBU_BUSCON0	Description
Boot memory data width	CFG16 CFG32	PORTW	8, 16, 32 Bit
Address generation control	AGEN	AGEN	multiplexed mode demultiplexed mode
Read Wait-State control	WAITRDC[6:5] WAITRDC[4:3]	WAITRDC	Number of Wait-States in read access (0..127)
Cycle Multiplier Control	CMULT CMULTR	CMULT	Wait cycle multiplier
		CMULTR	Read cycle multiplier
Byte Control mode	BCGEN	BCGEN	Functionality of BC[3:0]
Variable Wait-State insertion control	WAIT	WAIT	asynchronous synchronous
Extended Setup control	SETUP	SETUP	Cycle 0 generation
Active wait level control	WAITINV	WAITINV	$\overline{WAIT}$ = high or low

Table 11 Boot configuration register

Number of READ Wait states: nWS_R = CMULTR * WAITRDC
with CMULTR = *(1)*, 4, 8, 16 and WAITRDC = 0..127 (*48*)

Number of WRITE Wait states: nWS_W = CMULT * WAITWRC
with CMULT = 1, 4, 8, (*16)* and WAITWRC = 0..127 (*7*) (*defaults*)

To select “External boot as master via FPI Bus”, the boot configuration input pins of the TC1775 microcontroller should be connected as follows:

OCDSE = 1, $\overline{\text{BRKIN}}$ = 1, CFG[3:0] = 1101_B (1=VCC, 0=GND)

The PC Start value is set to 0xA0000000 (Segment 10).

If “External boot directly via EBU” is selected (CFG[3:0] = 1100_B), the PMU is connected directly to the EBU and performs burst mode cycles for an external code Flash memory. In this case, the reset value of $0000005F_H$ in the PMU External Instruction Fetch Control Register **PMU_EIFCON** will be used for the burst mode settings. The reset (default) values are defined as Burst length = 1 access in field **EIFBLEN**, two data cycles in field **DATLEN**, two address cycles in field **ADVLEN** within the initial address cycle and seven read wait cycles between the initial address cycle and the first instruction cycle in Field **RDWLEN**.

5.2 Burst mode configuration

The Program Memory Unit (PMU) of the TC1775 is designed to perform burst mode cycles to operate together with external code Flash memory in Burst Mode. For Burst Mode access to an external Flash memory device the PMU is directly connected to the External Bus Unit (EBU), which controls the connection to the external Flash device. Some types of Burst flash memory devices support the continous burst mode.

The functionality of continous instructions fetching used by the continous burst mode is **not implemented** in the TC1775, although this mode is supported by different Burst flash memory devices. For external burst flash mode **only 32-bit data bus width** is supported.

In the External Instruction Fetch Control register **PMU_EIFCON** the parameter **EIFBLEN** and **FFBLEN** are specified to define the burst length for an external burst request to the Flash memory. The burst length driven on the data bus depends on the specified burst buffer length so the minimum of specified burst length and burst buffer length will be used as the actual burst length.

A recommendation to define the burst length for different configurations is to set the value for the Flash Burst Buffer Length (FBBLEN) to 8 linear burst cycles and specify the actual burst length using the parameter External Instruction Flash Burst Length (EIFBLEN) with 1, 2, 4 or 8 data accesses.

Table 12 Actual Burst length definition

The burst flash memories use a n-Bit counter (e.g. Am29BL162CB-65CI uses a 5-Bit counter 0..31) to increment the address for the next data cycle during a burst starting with the initial address. If an overflow of this counter (...28-29-30-31-0-1-2-3...) occurs, the flash memory creates a signal which can be used by the microcontroller.

Burst length	EIFBLEN (Burst Length)	FBBLEN (Buffer Length)
2	01_B = 2 data accesses	010_B = 8 linear burst data cycles
4	10_B = 4 data accesses	
8	11_B = 8 data accesses	
2	01_B = 2 data accesses	001_B = 4 linear burst data cycles
4	10_B = 4 data accesses	

In case of a instruction cache miss each burst access to the external Flash memory begins at the missed address. The alignment of a cache line results to an 8-word address line border (address bits A[4:0] = 0).

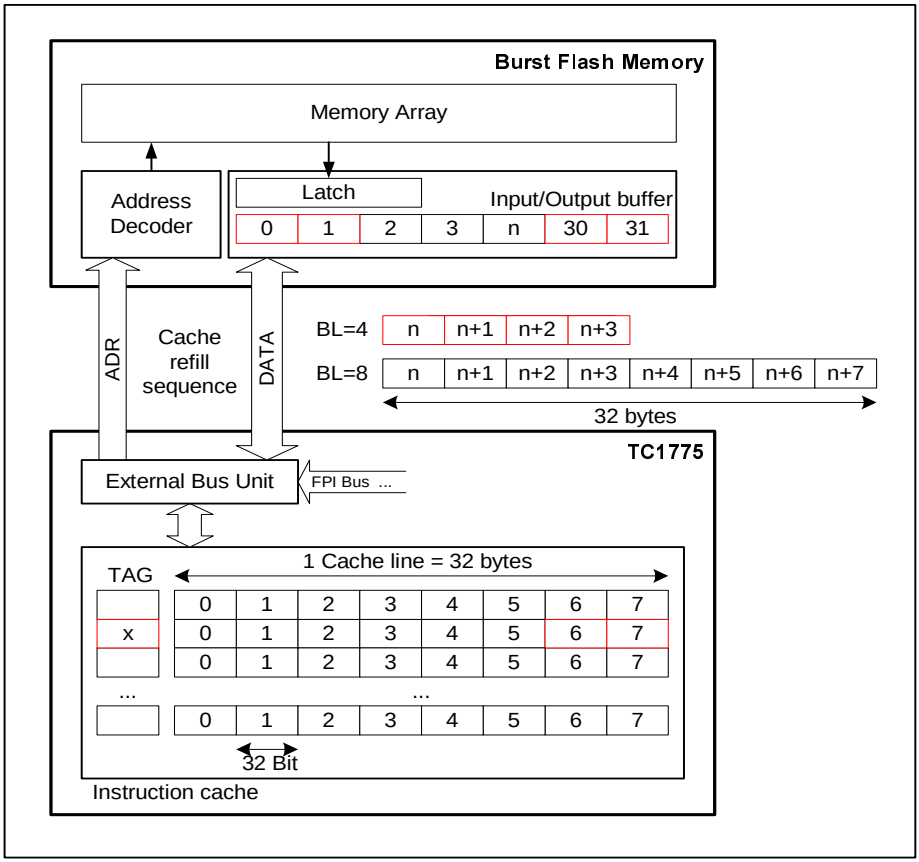

Figure 8 Cache refill sequence

For the TC1775 each cache line is aligned to A[4:0] = 0, so the offset is defined to 0x00, 0x20, 0x40, 0x60,... Every Cache refill sequence **starts with the missed address**. If a counter overflow occurs (30, 31, 0, 1), the data stream transferred from the Flash memory is not a continous address range. In the example above, only word 6 and 7 will be updated by word 30 and 31 by the content of the flash memory buffer. The words 0 and 1 will be received by the EBU but ignored to update any cache content.

Example:

Cache Miss for instruction at address 0xA0000078. The address lines A[21:2] of the EBU interface are connected to A[19:0] of the Flash memory, A[4:0] are connected inside the Flash memory to the burst counter.

BL=4 Offset=0x78 011110xx$_B$ Start counter=30; Burst order=30-31-0-1
BL=8 Offset=0x78 011110xx$_B$ Start counter=30; Burst order=30-31-0-1-2-3-4-5

5.2.1 Burst control register

Register	Name	Description
SCU_CON	EXTIF	1 = Instruction fetch directly (PMU <-> EBU)
	ENSWIF	1 = Enable changing of EXTIF bit
	EBUEN	1 = Enable EBU
PMU_EIFCON	ADVLEN	Number of address cycles (1..2)
	RDWLEN	Read wait cycles between initial address cycle and first data cycle (0..7)
	DATLEN	Number of data cycles (1..2)
	FBBMSEL	0 = ignore FBBLEN and EIFBLEN, a new request is starting at MISSED address up to the end of the cache line. 1 = Burst buffer length defined by value in FBBLEN
	FBBLEN	Maximum number of linear burst data cycles 001$_B$ = 4 010$_B$ = 8 011$_B$ = 16 100$_B$ = 32
PMU_EIFCON	EIFBLEN	Instruction Fetch Burst length 00$_B$ = 1 data access 01$_B$ = 2 data accesses 10$_B$ = 4 data accesses 11$_B$ = 8 data accesses

Table 13 Burst Flash configuration parameter

If EIFBLEN is set to 11$_B$, a configuration of FBBLEN = 001$_B$ will enable four external instruction fetch accesses (burst length = 4). The burst length depends on the setting of field FBBLEN. A new burst starts always at the MISSED address and goes until the specified parameters in EIFBLEN and FBBLEN if FBBMSEL (Mode select) is set to 1.

To guarantee correct functionality of the burst mode settings, it is recommended to set Bit 12 of the register PMU_EIFCON always to 0.

5.2.2 Synchronous burst read timing

Example:

ADVLEN = 0	One address cycle
RDWLEN = 1	One additional wait cycles between initial address cycle and first data
DATLEN = 0	One data cycle
FBBMSEL = 1	Flash burst buffer length defined by FBBLEN
FBBLEN = 010_B	Maximum 8 linear flash burst data cycles
EIFBLEN = 10_B	Instruction Fetch burst length = 4 data accesses

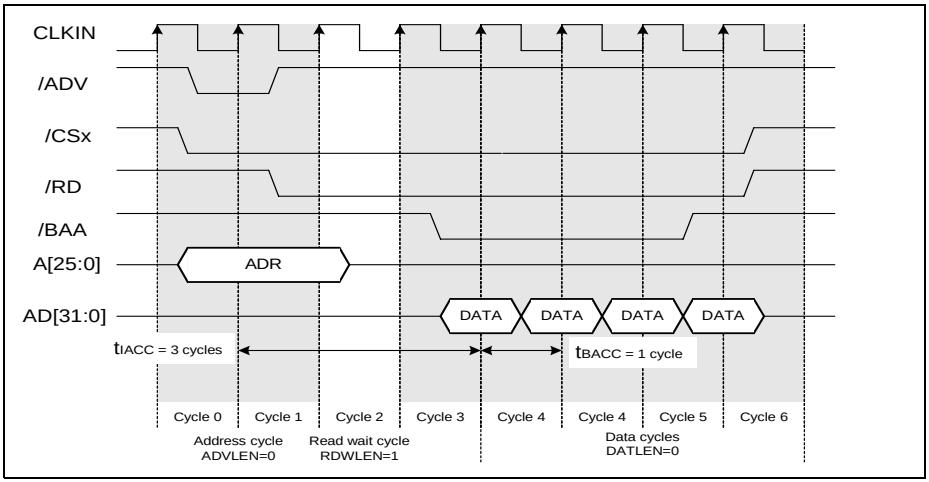

Figure 9 Synchronous burst read timing 3-1-1-1

*Note: The burst timing is related to the rising edge of the **CLKIN** signal*

Table 14 Synchronous burst read timing

Cycle	Description	Cycles min.	Cycles max.	Parameter
Cycle 0	Address cycle 1	1	2	PMU_EIFCON.ADVLEN
Cycle 1	Address cycle 2	0	1	PMU_EIFCON.SIDC
Cycle 2	Read wait cycle	0	7	PMU_EIFCON.RDWLEN
Cycle 3	Initial data cycle	1	1	fixed
Cycle 4	Burst data cycle	1	2	PMU_EIFCON.DATLEN
Cycle 5	Last burst data cycle	1	2	PMU_EIFCON.DATLEN
Cycle 6	End-of-burst cycle	1	1	fixed

6 Connecting memories to the TC1775 EBU

6.1 Asynchronous SRAM devices

Example:

2 x SRAM, 4 MBit 256k x 16, Alliance AS734098-15TI or SAMSUNG K6R4016V1C-15

The K6R4016V1C-15 is an asynchronous SRAM in x16 organization. It operates at a power supply of 3.3 Volt and uses 16 common input and output lines which can be controlled by two data byte control pins for upper and lower byte (DQ[15:8] and DQ[7:0]). The SRAM devices use an asynchronous, non-multiplexed address/data bus.

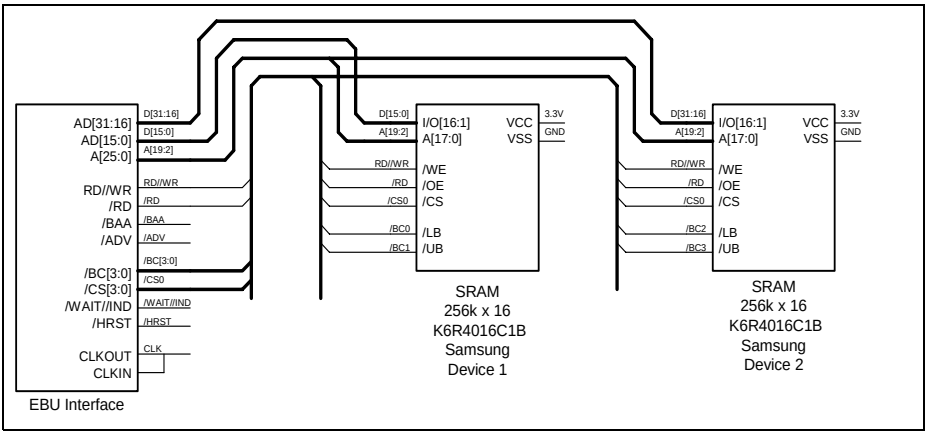

Figure 10 Interface to Samsung asynchronous SRAM, 32-Bit bus width

Total memory space = 2 * 4 MBit = 1 MByte

6.2 Connections between EBU interface and external memory

Pin name µC	Pin name Memory	Description
A[19:2]	A[17:0]	Address bus
AD[31:16]	DQ[15:0]	Data Input/Output, 16 bit, Device 1
AD[15:0]	DQ[15:0]	Data Input/Output, 16 bit, Device 2
$\overline{\text{CS0}}$	/CS	Chip Select (active low)
$\overline{\text{RD}}$	/OE	Output Enable, read Strobe (active low)
RD/$\overline{\text{WR}}$	/WE	Write Enable, write Strobe (active low)
$\overline{\text{BC0}}$	/LB	Byte Control, lower byte, Device 1 (active low)
$\overline{\text{BC1}}$	/HB	Byte Control, higher byte, Device 1 (active low)
$\overline{\text{BC2}}$	/LB	Byte Control, lower byte, Device 2 (active low)
$\overline{\text{BC3}}$	/HB	Byte Control, higher byte, Device 2 (active low)
3.3 V	VCC	Power Supply
GND	VSS	Power Supply

Table 15 Asynchronous SRAM configuration

The SRAM devices need Byte Control Input Signals for the different operating modes and also for selecting upper and lower output pins during read/write operations. The TriCore EBU supports external devices with a data width of 8, 16 and 32 bits and provides the Byte Control lines $\overline{\text{BC}}$[3:0]. The Byte Control lines are not generated if external memory is accessed directly via EBU and not via the FPI Bus. If Directly Instruction fetch is selected in the SCU configuration register SCU_CON.EXTIF or Directly Boot from external memory via EBU is selected by Boot options, the Program Memory Unit fetches instructions directly without using the FPI Bus. In this mode the EBU generates no Byte Control signals ($\overline{\text{BC}}$[3:0] = high) and activates the Code fetch status output $\overline{\text{CODE}}$. An external code fetch is not possible from SRAM during Direct Code Fetch Mode if the Byte Control signals $\overline{\text{BC}}$[3:0] are connected directly to the external SRAM. An AND-operation of $\overline{\text{BC}}$x and the $\overline{\text{CODE}}$ signals creates a Byte Control signal to have the opportunity to fetch code from external SRAM within Direct access mode without FPI Bus transfers (see **Figure 11**).

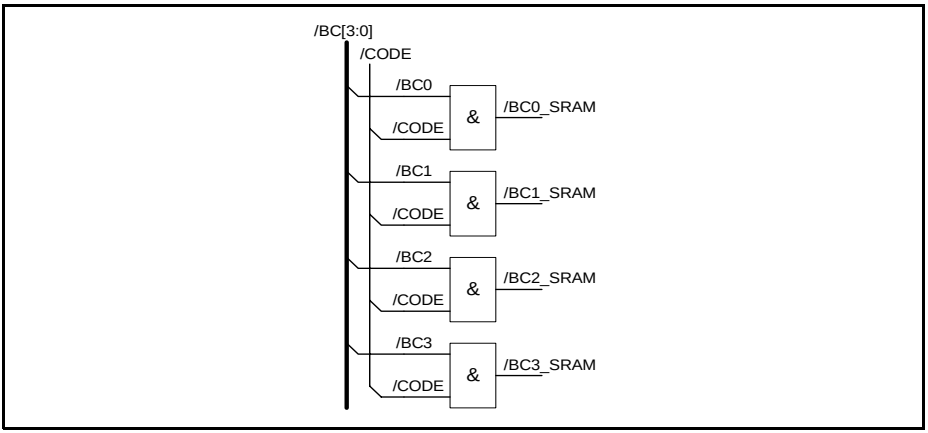

Figure 11 Generating a Byte Control signal to fetch code from external SRAM

6.2.1 Configuration

Definitions:

CPU clock: f_{SYS} = 40MHz, t_{SYS} = t_{CYCLE} = 25ns
SRAM write/read timing: t_{RC} = 15ns , t_{WC} = 15ns
Base address = 0xA0000000, SRAMs connected to $\overline{CS0}$
SRAM in asynchronous mode, 32-Bit bus width (2 x 16-Bit)
0 read Wait states, 0 write Wait states, 0 hold cycles, 0 recovery cycles

Address range:

4MBit + 4MBit = 8MBit = **1 MByte** (256k x 32)
A[26:20] will be compared to EBU_ADDSEL0.BASE
A[19:0] will be used to address memory within 1 Mbyte address range
EBU_BUSCON0.MASK = 0111_B, 7 address bits used for address comparison.

```
// SRAM address range = 0xA0000000 .. 0xA00FFFFF
// setting EBU_BUSCON: Bus configuration register
psEBU = (EBU *)(EBUA_BASE);      // pointer to EBU structure
psEBU->EBU_CON=0x0000FF68;       // Time-Out = 0xFF x 8 clock cycles
                                 // EBU=ext. Master, Ext.Access to FPI Bus
                                 // AGEN=0 -> demultiplexed mode
psEBU->EBU_BUSCON0=0x00020000;   // 32Bit, No Waitstates,
                                 // No hold + recovery cycles
psEBU->EBU_ADDSEL0=0xA0000071;   // Enable region 0, Mask=7,
                                 // Base = 0xA0000000
```

Note: It is recommended to make the setup of the External Bus Configuration in register EBU_BUSCONx before the address range will be enabled in register EBU_ADDSELx

6.3 Asynchronous Flash memory devices

Example: 2 x Flash memory, 16 MBit 1M x 16, AMD, Am29LV160BA-70

The Am29LV160BA is a 16 MBit, 3.3 Volt-only Flash memory device organized as 1Mx16. The device operates as a asynchronous Flash EPROM using the standard control pins WE#, OE# and CE#. This device is designed to be programmed in-system with the standard system 3.0 Volt V_{CC} supply. A 12.0 Volt V_{PP} or 5.0 V_{CC} are not required for write or erase operations. The device requires only a **single 3.3 Volt power supply** for both read and write functions. Internally generated and regulated voltages are provided for the program and erase operations.

The device offers access times of 70, 80, 90, and 120 ns.

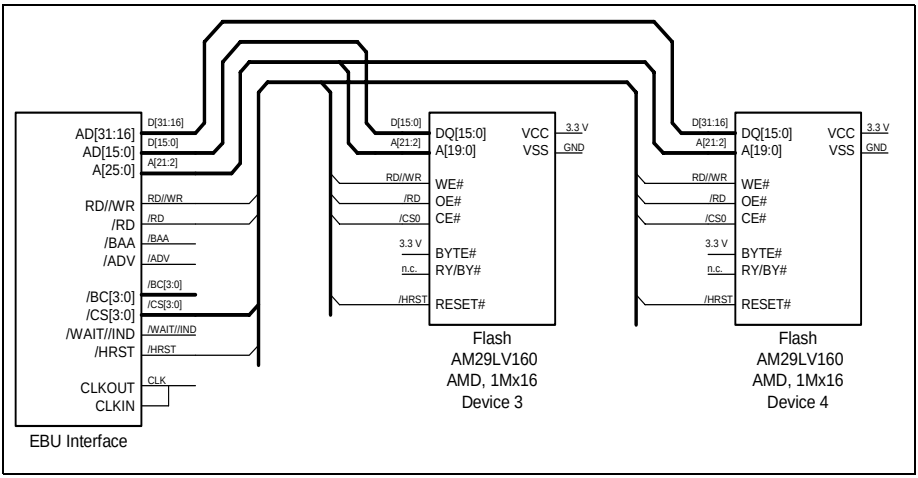

Figure 12 Interface to asynchronous AMD Flash devices, 32-Bit bus width

Total memory space = 2 * 16 MBit = 4 MByte

6.3.1 Connections between EBU interface and external memory

Pin name µC	Pin name Memory	Description
A[21:2]	A[19:0]	Address bus
AD[31:16]	DQ[15:0]	Data Input/Output, 16 bit, Device 3
AD[15:0]	DQ[15:0]	Data Input/Output, 16 bit, Device 4
$\overline{\text{CS0}}$	CS#	Chip Select (active low)
$\overline{\text{RD}}$	OE#	Output Enable, read Strobe (active low)
RD/$\overline{\text{WR}}$	WE#	Write Enable, write Strobe (active low)
3.3 V	BYTE#	Select 8-Bit or 16-Bit mode (high 16-Bit mode)
N.C.	RY/BY#	Ready busy (active low)
$\overline{\text{HRST}}$	RESET#	Reset signal (active low)
3.3 V	VCC	Power Supply
GND	VSS	Power Supply

Table 16 Asynchronous Flash memory configuration

6.3.2 Configuration

Definitions:

CPU clock: f_{SYS} = 40MHz, t_{SYS} = t_{CYCLE} = 25ns
Flash write/read timing: t_{ACC}=70ns, t_{RC}=70ns, t_{WC}=70ns
Base address = 0xA0000000, Flash devices connected to $\overline{CS0}$
Flash in asynchronous mode, 32-Bit bus width (2 x 16-Bit),
2 read Wait states, 2 write Wait states, 0 hold cycles, 0 recovery cycles

Address range:

16MBit + 16MBit = 32MBit = **4 MByte** (1M x 32)
A[26:20] will be compared to EBU_ADDSEL0.BASE
A[21:0] will be used to address memory within 4 Mbyte address range
EBU_BUSCON0.MASK = 0101_B, 5 address bits used for address comparison
EBU_BUSCON0.CMULT = 00_B Multiplier = 1
EBU_BUSCON0.WAITWRC = 2
EBU_BUSCON0.CMULTR = 00_B Multiplier = 1
EBU_BUSCON0.WAITRDC = 2
EBU_BUSCON0.RECOVC = 00_B No recovery cycle
EBU_BUSCON0.HOLDC = 00_B No Hold cycle

```
// Flash address range = 0xA0000000 .. 0xA03FFFFF
// setting EBU_BUSCON : Bus configuration register
psEBU = (EBU *)(EBUA_BASE);      // pointer to EBU structure
psEBU->EBU_CON=0x0000FF68;       // Time-Out = 0xFF x 8 clock cycles
                                 // EBU=ext. Master, Ext.Access to FPI Bus
                                 // AGEN=0 -> demultiplexed mode

// WAITWRC=2, CMULT=0, Multiplier=1, WAITRDC=2, CMULTR=0, Multiplier=1
psEBU->EBU_BUSCON0=0x00020480;   // 32Bit, No hold + recovery cycles
psEBU->EBU_ADDSEL0=0xA0000051;   // Enable region 0, Mask=5,
                                 // Base = 0xA0000000
```

A second way to set the value of EBU_BUSCON0 can be done by calculation of the register value:

```
uiWaitStates=2;                  // set Wait states to 2
uiConfig=0x00020000;             // 32Bit, No hold + recovery cycles
uiConfig|=uiWaitStates<<6;       // WAITWRC=2, CMULT=0, Multiplier=1
uiConfig|=uiWaitStates<<9;       // WAITRDC=2, CMULTR=0, Multiplier=1
psEBU->BUSCON0=uiConfig;         // Set BUSCON register
```

6.4 SRAM and Flash memory devices in asynchronous mode

To connect both SRAM and Flash memory to the EBU interface, $\overline{CS0}$ and $\overline{CS1}$ will be used to enable the devices. Two separate address ranges within a Segment can be used for RAM and Flash memory. In this example both ranges are located within Segment 10, which is the cached Segment. This configuration is the combination of SRAM and Flash (see **Chapter 6.1 'Asynchronous SRAM devices"** and **Chapter 6.3 'Asynchronous Flash memory devices"**) with the difference, that the SRAM range is mapped to address 0xA4000000.

Type	Base_Adr	Size	Chip Select	Read WS	Write WS
FLASH	0xA0000000	4 MByte	$\overline{CS0}$	2	2
SRAM	0xA4000000	1 MByte	$\overline{CS1}$	0	0

Table 17 SRAM and Flash memory devices

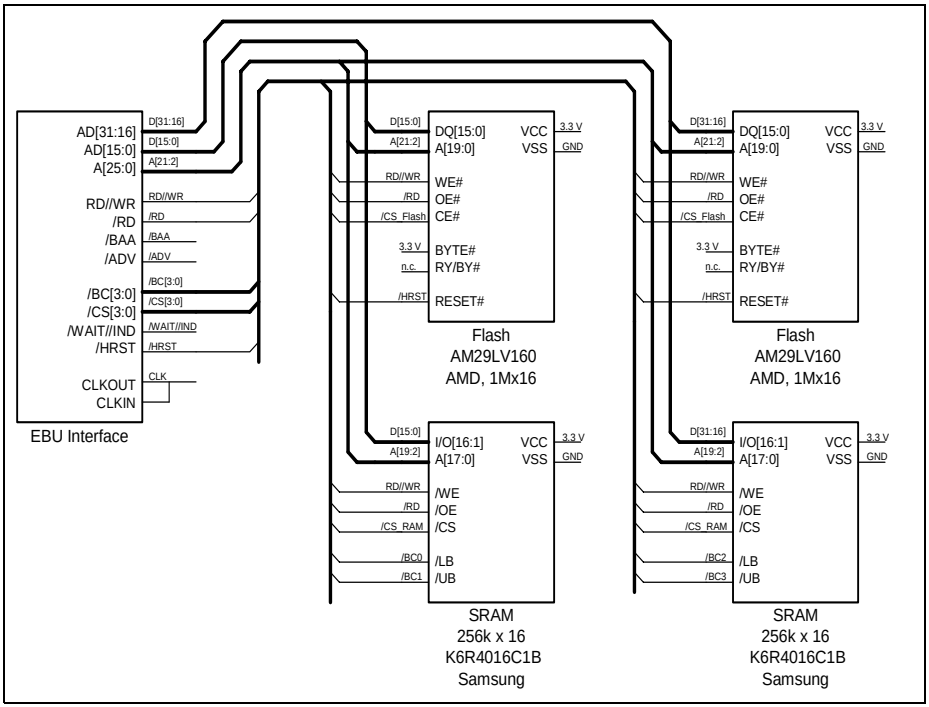

Figure 13 Interface to SRAM and Flash memory

6.4.1 Configuration

```
// Flash address range = 0xA0000000 .. 0xA03FFFFF  , 4 MByte, /CS0
// SRAM  address range = 0xA4000000 .. 0xA40FFFFF  , 1 MByte, /CS1
// setting EBU_BUSCON : Bus configuration register
psEBU = (EBU *)(EBUA_BASE);      // pointer to EBU structure
psEBU->EBU_CON = 0x0000FF68;     // Time-Out = 0xFF x 8 clock cycles
                                 // EBU=ext. Master,Ext.Access to FPI Bus
                                 // AGEN=0 -> demultiplexed mode
// Flash memory
// WAITWRC=2, CMULT=0, Multiplier=1, WAITRDC=2, CMULTR=0, Multiplier=1
psEBU->EBU_BUSCON0=0x00020480;   // 32Bit, No hold + recovery cycles
psEBU->EBU_ADDSEL0=0xA0000051;   // Enable region 0, Mask=5,
                                 // Base = psEBU->0xA0000000

// SRAM
// WAITWRC=0, CMULT=0, Multiplier=1, WAITRDC=0, CMULTR=0, Multiplier=1
psEBU->EBU_BUSCON1=0x00020000;   // 32Bit, No Waitstates,
                                 // No hold+recov. cycles
psEBU->EBU_ADDSEL1=0xA4000071;   // Enable region 1, Mask=7,
                                 // Base = 0xA4000000
```

A second way to set the value of EBU_BUSCON0 can be done by calculation of the register value :

```
uiWaitStates = 2;                // set Waitstates to 2
uiConfig = 0x00020000;           // 32Bit, No hold + recovery cycles
uiConfig |= uiWaitStates<<6;     // WAITWRC=2, CMULT=0, Multiplier=1
uiConfig |= uiWaitStates<<9;     // WAITRDC=2, CMULTR=0, Multiplier=1
psEBU->BUSCON0 = uiConfig;       // Set BUSCON register
```

This may be helpful if the application will be assembled with different types of Flash devices like different read/write access timing.

6.5 AMD Burst Flash memory devices

Example: 2 x Burst-Flash, 16 MBit 1M x 16, AMD, Am29BL162CB-65CI

The Am29BL162C is a 16 MBit, 3.0 Volt-only burst mode Flash memory device organized as 1Mx16. In burst mode the device allows a microcontroller to operate without wait states. The device can operate as a asynchronous Flash EPROM using the standard control pins WE#, OE# and CE#. After power-on the device starts in asynchronous read mode that allows the system to boot directly from the Flash. The User can set the operating mode to burst read mode using a burst mode enable command sequence (software sequence). In burst read mode, the data are transferred on the rising edge of the clock signal in combination with the load address pin $\overline{\text{LBA}}$ and the burst address advance pin $\overline{\text{BAA}}$.

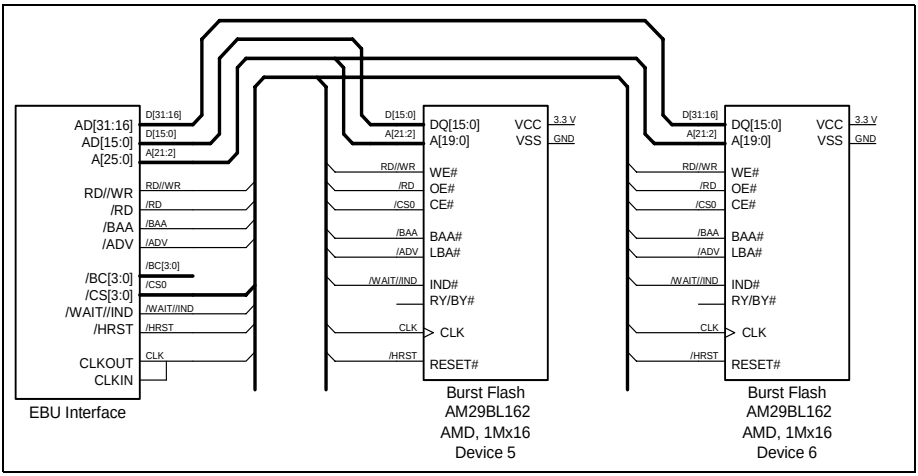

Figure 14 Interface to AMD Burst Flash devices, 32-Bit bus width

Total memory space = 2 * 16 MBit = 4 MByte

6.5.1 Connections between EBU interface and external memory

Table 18 AMD Burst Flash memory configuration

Pin name µC	Pin name Memory	Description
A[21:2]	A[19:0]	Address bus
AD[31:16]	DQ[15:0]	Data Input/Output, 16 bit, Device 5
AD[15:0]	DQ[15:0]	Data Input/Output, 16 bit, Device 6
$\overline{\text{CS0}}$	CS#	Chip Select (active low)
$\overline{\text{RD}}$	OE#	Output Enable, read Strobe (active low)
RD/$\overline{\text{WR}}$	WE#	Write Enable, write Strobe (active low)
CLKOUT	CLK	Clock
$\overline{\text{ADV}}$	LBA#	Load burst address (active low)
$\overline{\text{BAA}}$	BAA#	Burst address advance (active low)
$\overline{\text{WAIT}}$	IND#	Wait, Highest burst counter address reached (active low)
N.C.	RY/BY#	Ready busy (active low)
$\overline{\text{HRST}}$	RESET#	Reset signal (active low)
3.3 V	VCC	Power Supply
GND	VSS	Power Supply

6.5.2 Configuration

Definitions:

CPU clock: f_{SYS} = 40MHz, t_{SYS} = t_{CYCLE} = 25ns
Flash write/read timing: t_{ACC}=65ns, t_{RC}=65ns, t_{WC}=65ns, t_{IACC}=65ns, t_{BACC}=18ns
Base address = 0xA0000000, Flash devices connected to $\overline{CS0}$
Flash in asynchronous mode, 32-Bit bus width (2 x 16-Bit),
2 read Wait states, 2 write Wait states, 0 hold cycles, 0 recovery cycles

Address range:

16MBit + 16MBit = 32MBit = **4 MByte** (1M x 32)
A[26:20] will be compared to EBU_ADDSEL0.BASE
A[21:0] will be used to address memory within 4 Mbyte address range
EBU_BUSCON0.MASK = 0101_B, 5 address bits used for address comparison
EBU_BUSCON0.CMULT = 00_B Multiplier = 1
EBU_BUSCON0.WAITWRC = 2
EBU_BUSCON0.CMULTR = 00_B Multiplier = 1
EBU_BUSCON0.WAITRDC = 2
EBU_BUSCON0.RECOVC = 00_B No recovery cycle
EBU_BUSCON0.HOLDC = 00_B No Hold cycle

```
// Flash in asynchronous mode
// Flash address range = 0xA0000000 .. 0xA03FFFFF
// setting EBU_BUSCON : Bus configuration register
psEBU = (EBU *)(EBUA_BASE);      // pointer to EBU structure
psEBU->EBU_CON=0x0000FF68;       // Time-Out = 0xFF x 8 clock cycles
                                 // EBU=ext. Master, Ext.Access to FPI Bus
                                 // AGEN=0 -> demultiplexed mode

// Set 2 Read wait states, 2 Write wait states
// WAITWRC=2, CMULT=0, Multiplier=1, WAITRDC=2, CMULTR=0, Multiplier=1
psEBU->EBU_BUSCON0=0x00020480;   // 32Bit, No hold + recovery cycles
psEBU->EBU_ADDSEL0=0xA0000051;   // Enable region 0, Mask=5,
                                 // Base = 0xA0000000
```

A second way to set the value of EBU_BUSCON0 can be done by calculation of the register value:

```
uiWaitStates=2;                  // set Wait states to 2
uiConfig=0x00020000;             // 32Bit, No hold + recovery cycles
uiConfig|=uiWaitStates<<6;       // WAITWRC=2, CMULT=0, Multiplier=1
uiConfig|=uiWaitStates<<9;       // WAITRDC=2, CMULTR=0, Multiplier=1
psEBU->BUSCON0=uiConfig;         // Set BUSCON register
```

6.5.3 Flash burst mode

To execute code using the burst read mode on AMD Burst Flash devices (e.g. Am29BL162CB-65CI) both the Microcontroller and the Flash device must be initialized to perform burst read accesses. The Flash memory device starts in asynchronous read mode after power-on and expects a burst mode enable sequence to enter the burst mode.

```
// Address Offsets for configuration settings of AMD Flash memory
SETUP = 0x0555 * 4;        // AMD Setup Address offset for 32-Bit access
UNLOCK = 0x02AA * 4;       // AMD Unlock Address offset for 32-Bit access
Flash_Base = 0xA0000000;   // Flash base address
```

Cycle	CMD	Address	Data	Command
1	WR	Flash_Base + SETUP	0x00AA00AA	FLASH_SETUP
2	WR	Flash_Base + UNLOCK	0x00550055	FLASH_UNLOCK
3	WR	Flash_Base + SETUP	0x00C000C0	FLASH_BURST_MODE
4	WR	Flash_Base	0x00010001	FLASH_BURST_ENABLE
5	WR	Flash_Base	0x00F000F0	FLASH_RESET

Table 19 AMD Burst mode enable sequence based on 32-Bit data bus width

The external burst mode instruction fetches are controlled and defined by the PMU_EIFCON register which is located in the PMU. To switch the code fetch mode from asynchronous mode to synchronous burst mode, the code has to be executed from the internal or external SRAM. To enable the burst mode on both the external Flash memory and the PMU, the code has to run from a defined memory range that isn't located in external Flash memory. After the last FLASH_RESET command the Flash memory all instruction fetches are performed in the synchronous burst mode.

To guarantee that all data and instructions accesses to the system and local data memory busses will be successfully executed, at the end of the subroutine a data and instruction synchronization has to be performed using _isync() and _dsync() commands.

isync(): Forces completion of all previous instructions, flushes the CPU pipelines and invalidates any cached pipeline state before proceeding to the next instruction.

dsync(): Forces all data accesses to complete before any data accesses associated with an instruction semantically after the DSYNC are initiated.

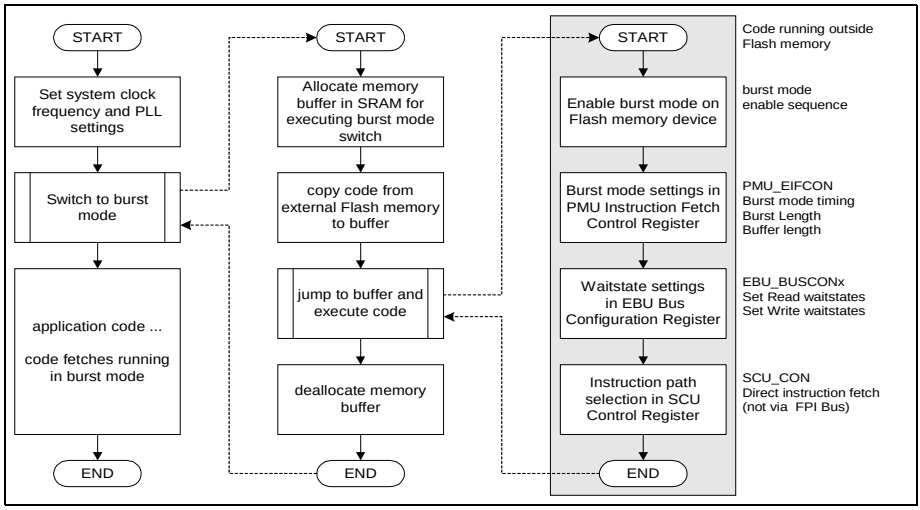

Figure 15 External burst mode instruction fetches

If external boot is selected, the TriCore starts execution of code from the external memory in asynchronous, demultiplexed address mode (see **"Booting from external memory" on Page 18**). To enable external burst mode instruction fetches, changes to the External Instruction Fetch Register (PMU_EIFCON), Bus configuration register (EBU_BUSCONx) and Control register of System Control Unit (SCU_CON) must be made.

Note: In some documents different interpretations of burst stream and Wait-State stream can be found. For the following examples the information 3-1-1-1 stands for burst stream and defines the number of clock cycles during the burst read access. The start of the burst is defined on the valid edge of the clock signal when a burst start address is valid.
burst = 3-1-1-1: 3 clock cycles initial access time, 1 clock cycle burst access time, No burst read Wait-States

```
// AMD Flash Timing for 40MHz clock frequency (NDIV=2)
// tIACC = 65ns, tBACC = 18ns, burst = 3-1-1-1
// Read timing (3 cycles) = cycle 1 + one Waitstate (RDWLEN=1)+ cycle 3
// one data cycle, one address cycle, burstlength = 8

psEBU = (EBU *)(EBUA_BASE);   // pointer to EBU structure
psSCU = (SCU *)(SCU_BASE);    // pointer to SCU structure
psPMU = (PMU *)(PMU_BASE);    // pointer to PMU structure

if(uiNDIV < 7)                // set Waitstates dependent on NDIV
  uiWaitstates = 2;           // timing for burst mode (uiWaitstate-1)
else                          // and asynchronous timing (uiWaitstate)
  uiWaitstates = 3;

// clearing Waitstates in EBU_BUSCON : Bus configuration register
uiConfig &= 0xFFFFFFFC;       // clear CMULT
uiConfig &= 0xFFFF003F;       // clear WAITRDC and WAITRDC
uiConfig &= 0xFF3FFFFF;       // clear CMULTR

// setting EBU_BUSCON : Bus configuration register
uiConfig |= uiWaitStates<<6;  // WAITWRC=n, CMULT=0, Multiplier=1
uiConfig |= uiWaitStates<<9;  // WAITRDC=n, CMULTR=0, Multiplier=1
psEBUA->BUSCON0 = uiConfig;   // set BUSCON register

// setting EIFCON : External instruction fetch register
uiConfig = (uiWaitStates-1)<<1;  // one additional read wait cycle
uiConfig |= PMUFBBMSEL;       // buffer length defined by FBBLEN
uiConfig |= PMUFBBLEN_8;      // Burst buffer length = 8
uiConfig |= PMUIFUBLEN_8;     // Instruction burst length = 8
psPMU->EIFCON = uiConfig;     // Set EIFCON register

// setting SCU_CON : System control unit
// Enable direct instruction fetch (not via FPI bus)
uiConfig = psSCU->CON;        // Load value of SCU
uiConfig |= ENSWIF;           // Enable switching of Instruct. Fetch Path
uiConfig |= EXTIF;            // Instruction fetch direct
uiConfig |= EBUEN;            // Enable EBU
psSCU->CON = uiConfig;

_isync();                     // synchronize instructions
_dsync();                     // synchronize data

// return to function call
```

6.6 Intel Burst Flash memory devices

Example: 2 x Burst-Flash, 16 MBit 1M x 16, Intel, 28F160F3B95

The Intel Fast Boot Block Flash memory offers highest performance using synchronous burst reads and supports asynchronous page mode operation for non-clocked memory subsystems. In synchronous burst mode, the CLK input increments an internal burst address generator, synchronizes the flash memory with the host CPU, and outputs data on every rising (or falling) CLK edge up to 60 MHz. Synchronous burst reads are enabled by configuring the read configuration register using the standard two-bus-cycle algorithm. The 16 MBit device is organized in x16 organization and operates on 3.3 Volt power supply.

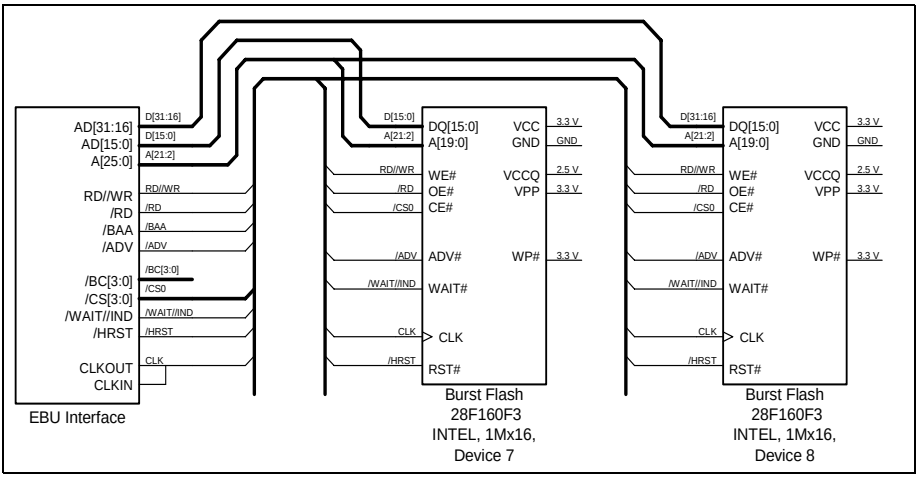

Figure 16 Interface to INTEL Burst Flash devices, 32-Bit bus width

Total memory space = 2 * 16 MBit = 4 MByte

6.6.1 Connections between EBU interface and external memory

Pin name µC	Pin name Memory	Description
A[21:2]	A[19:0]	Address bus
AD[31:16]	DQ[15:0]	Data Input/Output, 16 bit, Device 7
AD[15:0]	DQ[15:0]	Data Input/Output, 16 bit, Device 8
$\overline{CS0}$	CE#	Chip Enable (active low)
$\overline{RD}$	OE#	Output Enable, read Strobe (active low)
RD/$\overline{WR}$	WE#	Write Enable, write Strobe (active low)
CLKOUT	CLK	Clock
$\overline{ADV}$	ADV#	Address valid (active low)
$\overline{WAIT}$	WAIT#	Data valid feedback during burst mode (active low)
$\overline{HRST}$	RST#	Reset signal (active low)
3.3 V	WP#	Write protection (active low)
3.3 V	VCC	Power Supply
3.3 V	VPP	Block erase and program power supply
2.5 V	VCCQ	Flash Output buffer voltage
GND	GND	Power Supply

Table 20 Intel Burst Flash memory configuration

6.6.2 Configuration

Definitions:

CPU clock = 40MHz, t_{SYS} = 25ns, t_{ACC} = 90ns,

Intel recommendation: CPU clock < 50MHz => frequency configuration code = 3 (t_{IACC} = 4 clock cycles, t_{BACC} = 1 clock cycle, burst = 4-1-1-1)

Flash in asynchronous mode, 32-Bit bus width (2 x 16-Bit), 3 read Wait states, 3 write Wait states, 0 hold cycles, 0 recovery cycles, Flash devices connected to $\overline{CS0}$

Address range:

16MBit + 16MBit = 32MBit = **4 Mbyte** (1M x 32)
A[26:20] will be compared to EBU_ADDSE0L.BASE
A[21:0] will be used to address memory within 4 Mbyte address range
EBU_BUSCON0.MASK = 0101_B, 5 address bits
EBU_BUSCON0.CMULT = 00_B Multiplier = 1
EBU_BUSCON0.WAITWRC = 3
EBU_BUSCON0.CMULTR = 00_B Multiplier = 1
EBU_BUSCON0.WAITRDC = 3
EBU_BUSCON0.RECOVC = 00_B No recovery cycle
EBU_BUSCON0.HOLDC = 00_B No Hold cycle

```
// Flash in asynchronous mode
// Flash address range = 0xA0000000 .. 0xA03FFFFF
// setting EBU_BUSCON : Bus configuration register
psEBU = (EBU *)(EBUA_BASE);      // pointer to EBU structure
psEBU->EBU_CON=0x0000FF68;       // Time-Out = 0xFF x 8 clock cycles
                                 // EBU=ext. Master, Ext.Access to FPI Bus
                                 // AGEN=0 -> demultiplexed mode

// Set 3 Read wait states, 3 Write wait states
// WAITWRC=3, CMULT=0, Multiplier=1, WAITRDC=3, CMULTR=0, Multiplier=1
psEBU->EBU_BUSCON0=0x000206C0;   // 32Bit, No hold + recovery cycles
psEBU->EBU_ADDSEL0=0xA0000051;   // Enable region 0, Mask=5,
                                 // Base = 0xA0000000
```

A second way to set the value of EBU_BUSCON0 can be done by calculation of the register value:

```
uiWaitStates=3;                  // set Wait states to 3
uiConfig=0x00020000;             // 32Bit, No hold + recovery cycles
uiConfig|=uiWaitStates<<6;       // WAITWRC=3, CMULT=0, Multiplier=1
uiConfig|=uiWaitStates<<9;       // WAITRDC=3, CMULTR=0, Multiplier=1
psEBU->BUSCON0=uiConfig;         // Set BUSCON register
```

6.6.3 Flash burst mode

To execute code using the burst read mode on Intel Burst Flash devices (e.g. 28F160F3B95), the Flash device must be initialized to perform burst read accesses. The Flash memory device starts in asynchronous read mode after power-on and expects a configuration data word in the Read Configuration Command register to enter the burst mode.

The Set Read Configuration command writes data to the read configuration register. This operation is initiated by a standard two bus cycle command sequence. The Read Configuration Setup command (60_H) is written to an address within the Flash device, where Bits[15:0] include the configuration value. A second write with command 03_H confirms the operation where also the configuration value has to be part of the address.

```
RCD = RCD * 4;              // INTEL Configuration cmd value for 32-Bit
Flash_Base = 0xA0000000;    // Flash base address
```

Cycle	CMD	Address	Data	Command
1	WR	Flash_Base + RCD	0x00600060	SET RCD CYCLE#1
2	WR	Flash_Base + RCD	0x00030003	SET RCD CYCLE#2
3	WR	Flash_Base	0x00FF00FF	READ ARRAY RESET

Table 21 Intel Set Read Configuration Command sequence (32-Bit access)

The 16-Bit read configuration register value has to be placed on the address bus, A[15:0]. Two configuration cycles are necessary to set the value of the read configuration register.

The frequency configuration code specifies the number of initial data access cycles (t_{IACC}). For the Burst Flash devices 28F160F3B Intel specifies

CODE 3 for -95 ns access time, VCC=3.3V, 40MHz < clock frequency <= 50MHz

and

CODE 4 for -120 ns access time, VCC=3.3V, clock frequency <= 46MHz

Name	Field	Bit(s)	Description
Burst length BL	RCR[2:0]	3	001b = 4 Word Burst 010b = 8 Word Burst 111b = Continuous Burst
Clock configuration CC	RCR[6]	1	0 = falling edge 1 = rising edge
Burst sequence BS	RCR[7]	1	0 = Intel burst order 1 = linear order
Data Output Config DOC	RCR[9]	1	0 = one data hold cycle 1 = two data hold cycles
Frequency configuration	RCR[13:11]	3	binary value for CODE 011b = CODE 3 (4 clock cycles) 100b = CODE 4 (5 clock cycles)
Read mode RM	RCR[15]	1	0 = enable burst mode

Table 22 Specification of Intel Read Configuration register

```
// Initialization code for Intel Burst Flash

uint32 *puiAdr;               // Address pointer
uiWaitstates = 3;             // NDiv = 2, 40 MHz, value for CODE
uiFlashBase = 0xA0000000;     // Address range = 0xA0000000 .. 0xA03FFFFF

uiRCRvalue = uiWaitstates<<11 | 0x00C2;    // lin.burst,BL=8,rising edge

puiAdr = (uint32*) (uiFlashBase + uiRCRvalue*4);// 32-Bit calculation

*puiAdr = 0x00600060;         // Set Read configuration cycle#1
*puiAdr = 0x00030003;         // Set Read configuration cycle#2
*puiAdr = 0x00FF00FF;         // Set Read Array Reset
```

```
// INTEL Flash Timing for 40MHz clock frequency (NDIV=2)
// tIACC = 95ns, output data on every rising or falling edge up to 65 MHz
// burst = 4-1-1-1
// Read timing (4 cycles) = cycle 1 + two Waitstate (RDWLEN=2) + cycle 3
// one data cycle, one address cycle, burstlength = 8

psEBU = (EBU *)(EBUA_BASE);   // pointer to EBU structure
psSCU = (SCU *)(SCU_BASE);    // pointer to SCU structure
psPMU = (PMU *)(PMU_BASE);    // pointer to PMU structure

if(uiNDIV > 5)                // set Waitstates dependent on NDIV
  uiWaitstates = 4;           // Waitstates for asynchronous timing
else
  uiWaitstates = 3;

// clearing Waitstates in EBU_BUSCON : Bus configuration register
uiConfig &= 0xFFFFFFFC;       // clear CMULT
uiConfig &= 0xFFFF003F;       // clear WAITRDC and WAITRDC
uiConfig &= 0xFF3FFFFF;       // clear CMULTR

// setting EBU_BUSCON : Bus configuration register
uiConfig |= uiWaitStates<<6;  // WAITWRC=n, CMULT=0, Multiplier=1
uiConfig |= uiWaitStates<<9;  // WAITRDC=n, CMULTR=0, Multiplier=1
psEBUA->BUSCON0 = uiConfig;   // set BUSCON register

// setting EIFCON : External instruction fetch register
uiConfig = (uiWaitStates-1)<<1; // one additional read wait cycle
uiConfig |= PMUFBBMSEL;       // buffer length defined by FBBLEN
uiConfig |= PMUFBBLEN_8;      // Burst buffer length = 8
uiConfig |= PMUIFUBLEN_8;     // Instruction burst length = 8
psPMU->EIFCON = uiConfig;     // Set EIFCON register

// setting SCU_CON : System control unit
// Enable direct instruction fetch (not via FPI bus)
uiConfig = psSCU->CON;        // Load value of SCU
uiConfig |= ENSWIF;           // Enable switching of Instruct. Fetch Path
uiConfig |= EXTIF;            // Instruction fetch direct
uiConfig |= EBUEN;            // Enable EBU
psSCU->CON = uiConfig;

_isync();                     // synchronize instructions
_dsync();                     // synchronize data

// return to function call
```

6.7 ST Microelectronics Burst Flash memory devices

Example: 1 x Burst-Flash, 16 MBit 512k x 32, ST, M58BW016

The M58BW016B/D is 3V Flash Memory in x32 organization. An optional VPP of 12 Volt can be provided to speed-up program and erase operations. This Flash memory device has separate VDDQ and VDDQIN power supply pins for I/O buffers. This is useful to interface the Flash Memory device with lower or higher power supply devices. VDDQ(IN) can go from 2.4V to VDD (if VDD=3.3 Volt, VDDQ can be between 2.4 Volt and 3 Volt). A power supply voltage of 3.3 Volt (2.7 Volt minimum) is mandatory to drive the core circuits of the Flash Memory.

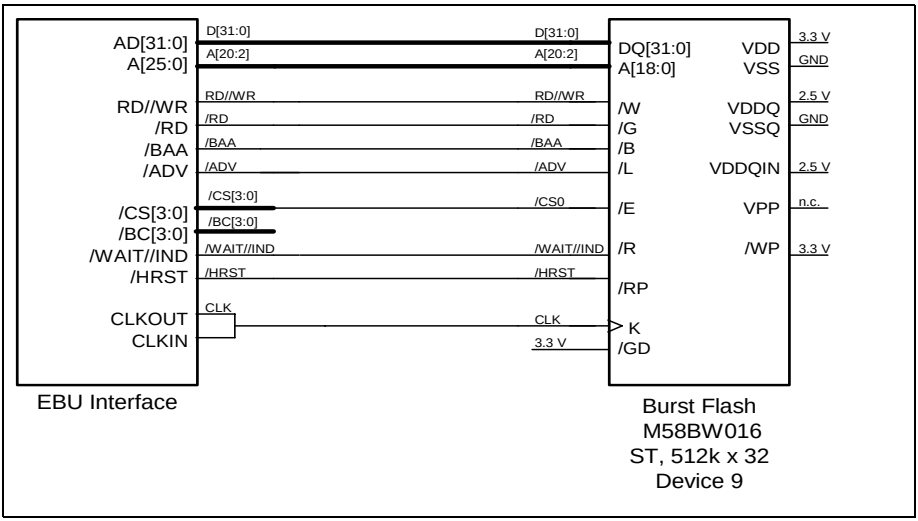

Figure 17 Interface to ST Burst Flash devices, 32-Bit bus width

Total memory space = 1 * 16 MBit = 2 MByte

6.7.1 Connections between EBU interface and external memory

Table 23 ST Burst Flash memory configuration

Pin name µC	Pin name Memory	Description
A[20:2]	A[18:0]	Address bus
AD[31:0]	DQ[31:0]	Data Input/Output, 32 bit, Device 9
$\overline{CS0}$	/E	Chip Select, Chip Enable (active low)
$\overline{RD}$	/G	Output Enable, read Strobe (active low), low when the Flash Memory is in read mode, high during a Flash write operation.
3.3 V	/GD	Output Disable (active low), deactivates the output buffers and set to high impedance, must be connected to an external pull-up resistor
RD/$\overline{WR}$	/W	Write Enable, write Strobe (active low)
$\overline{BAA}$	/B	Burst address advance (active low)
$\overline{ADV}$	/L	Latch Enable (active low), Load burst address (active low)
CLKOUT	K	Burst Clock
N.C.	R	Valid data ready, Open Drain output, identifies if the memory is ready to output data
3.3 V	/WP	Write protect
$\overline{HRST}$	/RP	Reset/Power down (active low)
3.3 V	VDD	Supply voltage
2.5 V	VDDQ	Flash Output supply voltage, used for DQ's
2.5 V	VDDQIN	Flash Input supply voltage, used for all input signals
3.3 V	VPP	Program/Erase Supply Voltage
GND	VSS	Ground, Power Supply
GND	VSSQ	Ground, Output supply voltage

Asynchronous Read is the default read mode which the Flash device enters on power-up or on return from Reset/Power-Down. An asynchronous read cycle is performed when detecting a valid address on the address inputs A[18:0], Chip Enable (/E) = low, Output Enable (/G) = low, Write Enable (/W) = high and Output Disable (/GD) = high.

Synchronous Burst Read mode may be enabled after executing a initialization sequence (**Chapter 6.7.3 'Flash burst mode"**). A valid Synchronous Burst Read operation begins when the Burst Clock (K) is active and Chip Enable (/E) and Latch Enable (/L) are low. The burst start address (A[18:0]) is latched and loaded into the internal Burst Address Counter on the valid Burst Clock (K) edge or on the rising edge of Latch Enable, whichever occurs first.

After an initial burst access time t_{ACI}, the memory may output data on each clock cycle for the burst configuration X-1-1-1 depending on the clock frequency. The Burst Address Advance input (/B) controls the memory burst output.

Note: In brackets the pin marking of the M58BW016B/D Flash memory device is described.

6.7.2 Configuration

CPU clock = 40MHz, t_{SYS} = 25ns,
Flash timing: t_{ACC} = 80ns, $t_{SETUP(MIN)}$ = 6ns, $t_{HOLD(MIN)}$ = 3ns
Flash in asynchronous mode, 32-Bit bus width, 3 read Wait states, 3 write Wait states, 0 hold cycles, 0 recovery cycles, Flash devices connected to $\overline{CS0}$

Base address = 0xA0000000, EBU is external master

16MBit = 2 Mbyte (512K x 32)
A[26:21] will be compared to EBU_ADDSE0L.BASE
A[20:0] will be used to address memory within 2 Mbyte address range
EBU_CONF0.MASK = 0110_B, 6 address bits
EBU_CONF0.CMULT = 00_B Multiplier = 1
EBU_CONF0.WAITWRC = 3
EBU_CONF0.CMULTR = 00_B Multiplier = 1
EBU_CONF0.WAITRDC = 3
EBU_CONF0.RECOVC = 00_B No recovery cycle
EBU_CONF0.HOLDC = 00_B No Hold cycle

```
// Flash address range = 0xA0000000 .. 0xA03FFFFF
// setting EBU_BUSCON : Bus configuration register
psEBU = (EBU *)(EBUA_BASE);   // pointer to EBU structure
uiWaitStates = 3;             // set Waitstates to 3
psEBU->EBU_CON = 0x0000FF68;  // Time-Out = 0xFF x 8 clock cycles
                              // EBU=ext. Master, Ext. Access to FPI Bus
                              // AGEN=0 -> demultiplexed mode
psEBU->EBU_ADDSEL0 = 0xA0000061; // Enable region 0, Mask=6,
                              // Base = 0xA0000000
uiConfig = 0x00020000;        // 32Bit, No hold + recovery cycles
uiConfig |= uiWaitStates<<6;  // WAITWRC=3, CMULT=0, Multiplier=1
uiConfig |= uiWaitStates<<9;  // WAITRDC=3, CMULTR=0, Multiplier=1
psEBU->BUSCON0 = uiConfig;    // Set BUSCON register
```

6.7.3 Flash burst mode

To execute code using the burst read mode on ST Burst Flash devices (e.g. M58BW016BT80) the Flash device must be initialized to perform burst read accesses. The Flash memory device starts in asynchronous read mode after power-on and expects a configuration data word in the Burst Configuration Register (BCR) to enter the burst mode. The burst mode initialization code has to run outside the Flash memory.

The Set Burst Configuration Register command writes data to the Burst configuration register. This operation is initiated by a standard two bus cycle command sequence. The Set Burst Configuration Register command 0x60 is written to any address within the Flash memory device to enter the procedure. In the second cycle the value 0x03 has to be written to a valid address within the Flash memory device where the Burst Configuration Word is transmitted on A[15:0].

```
BCR = 0x10CA;              // Value for Burst Configuration register
BCR = BCR * 4;             // ST Burst Configuration cmd value for 32-Bit
Flash_Base = 0xA0000000;   // Flash base address
```

Cycle	CMD	Address	Data	Command
1	WR	Flash_Base	0x00000060	SET RCD CYCLE#1
2	WR	Flash_Base + BCR	0x00030003	SET RCD CYCLE#2

Table 24 ST Set Burst Configuration Command sequence (32-Bit access)

```
// Initialization code for ST Burst Flash

uint32 *puiAdr;            // Address pointer
uiWaitstates = 4;          // NDiv=2, 40 MHz, value for X-Latency, e.g. 4
uiFlashBase = 0xA0000000;  // Address range = 0xA0000000 .. 0xA03FFFFF

uiBCRvalue = (uiWaitstates-2)<<11 | 0x00C2;// BL=8, ris. edge, one clock

puiAdr = (uint32*) (uiFlashBase);// Set Adr pointer to Flash base adr
*puiAdr = 0x00000060;            // Set Read configuration cycle#1
puiAdr = (uint32*) (uiFlashBase + uiBCRvalue*4) // 32-Bit adr calculation
*puiAdr = 0x00000003;            // Set Read configuration cycle#2
```

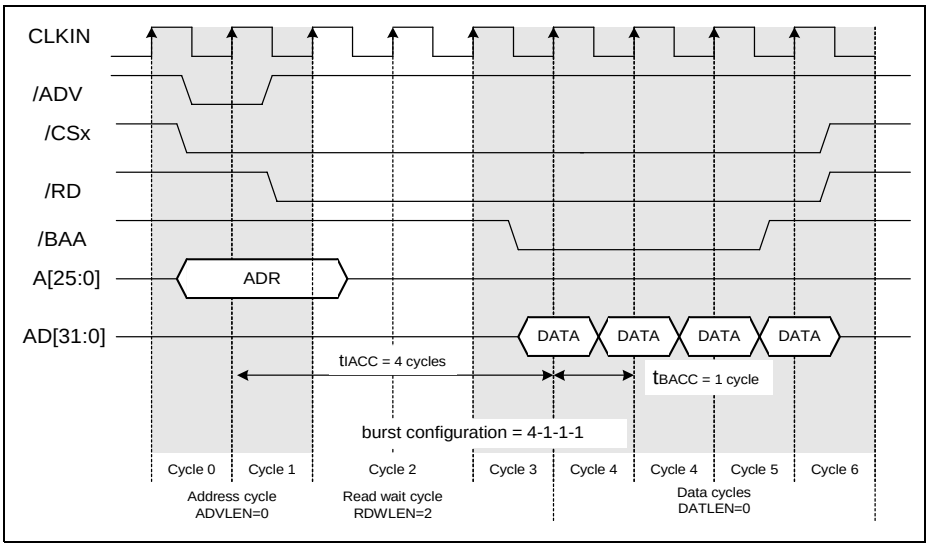

Figure 18 Burst configuration 4-1-1-1 example

Table 25 Specification of ST Burst Configuration register

If Y-Latency is set to 0, a data word (32-Bit) is driven on the data bus on each valid clock edge. The initial access time depends on the parameter X-Latency.

Name	Field	Bit(s)	Description
Burst length	BCR[2:0]	3	001b = 4 Word Burst 010b = 8 Word Burst 111b = Continuous Burst
Wrapping	BCR[3]	1	0 = boundary wrap 1 = no boundary wrap
Valid clock edge	BCR[6]	1	0 = falling edge 1 = rising edge
Burst Type	BCR[7]	1	0 = Interleaved burst order 1 = Sequential burst order
Valid data ready	BCR[8]	1	0 = Pin R is low during burst clock 1 = Pin R is low one data cycle before
Y-Latency	BCR[9]	1	0 = one burst clock cycle 1 = two burst clock cycles
X-Latency	BCR[13:11]	3	initial access time latency 010b = 4, 4-1-1-1 011b = 5, 5-1-1-1 or 5-2-2-2 100b = 6, 6-1-1-1 or 6-2-2-2
Read mode	BCR[15]	1	0 = synchronous burst read 1 = asynchronous read (default)

```
// ST Flash Timing for 40MHz clock frequency (NDIV=2)
// tIACC = 80ns, output data on every rising or falling edge up to 65 MHz
// burst = 4-1-1-1 (four intial clock cycles, one burst clock cycle)
// Read timing (4 cycles) = cycle 1 + two Wait states (RDWLEN=2) + cycle 3
// one data cycle, one address cycle, burstlength = 8

psEBU = (EBU *)(EBUA_BASE);   // pointer to EBU structure
psSCU = (SCU *)(SCU_BASE);    // pointer to SCU structure
psPMU = (PMU *)(PMU_BASE);    // pointer to PMU structure

if(uiNDIV > 5)                // set Waitstates dependent on NDIV
  uiWaitstates = 4;           // Waitstates for asynchronous timing
else
  uiWaitstates = 3;

// clearing Waitstates in EBU_BUSCON : Bus configuration register
uiConfig &= 0xFFFFFFFC;       // clear CMULT
uiConfig &= 0xFFFF003F;       // clear WAITRDC and WAITRDC
uiConfig &= 0xFF3FFFFF;       // clear CMULTR

// setting EBU_BUSCON : Bus configuration register
uiConfig |= uiWaitStates<<6;  // WAITWRC=n, CMULT=0, Multiplier=1
uiConfig |= uiWaitStates<<9;  // WAITRDC=n, CMULTR=0, Multiplier=1
psEBUA->BUSCON0 = uiConfig;   // set BUSCON register

// setting EIFCON : External instruction fetch register
uiConfig = (uiWaitStates-1)<<1; // one additional read wait cycle
uiConfig |= PMUFBBMSEL;       // buffer length defined by FBBLEN
uiConfig |= PMUFBBLEN_8;      // Burst buffer length = 8
uiConfig |= PMUIFUBLEN_8;     // Instruction burst length = 8
psPMU->EIFCON = uiConfig;     // Set EIFCON register

// setting SCU_CON : System control unit
// Enable direct instruction fetch (not via FPI bus)
uiConfig = psSCU->CON;        // Load value of SCU
uiConfig |= ENSWIF;           // Enable switching of Instruct. Fetch Path
uiConfig |= EXTIF;            // Instruction fetch direct
uiConfig |= EBUEN;            // Enable EBU
psSCU->CON = uiConfig;        // Set CON register

_isync();                     // synchronize instructions
_dsync();                     // synchronize data

// return to function call
```

6.8 TC1775 EBU and PMU settings

As described in **Chapter 4.4 'Read Access timing for demultiplexed mode"** and **Chapter 5.2 'Burst mode configuration"** , the register **EBU_BUSCONx** is used for the timing of external bus accesses in asynchronous mode and register PMU_EIFCONx is used for external instruction fetch control. To initialize the modes and functionalities for the external bus interface, the registers may be set directly with the configuration value or with a calculated value. Additional information to the functionality of each bit can be found in the TC1775 User's manual. The most important parameters for the timing on the external memory bus for asynchronous mode and synchronous burst mode can be found together with some examples in the table below.

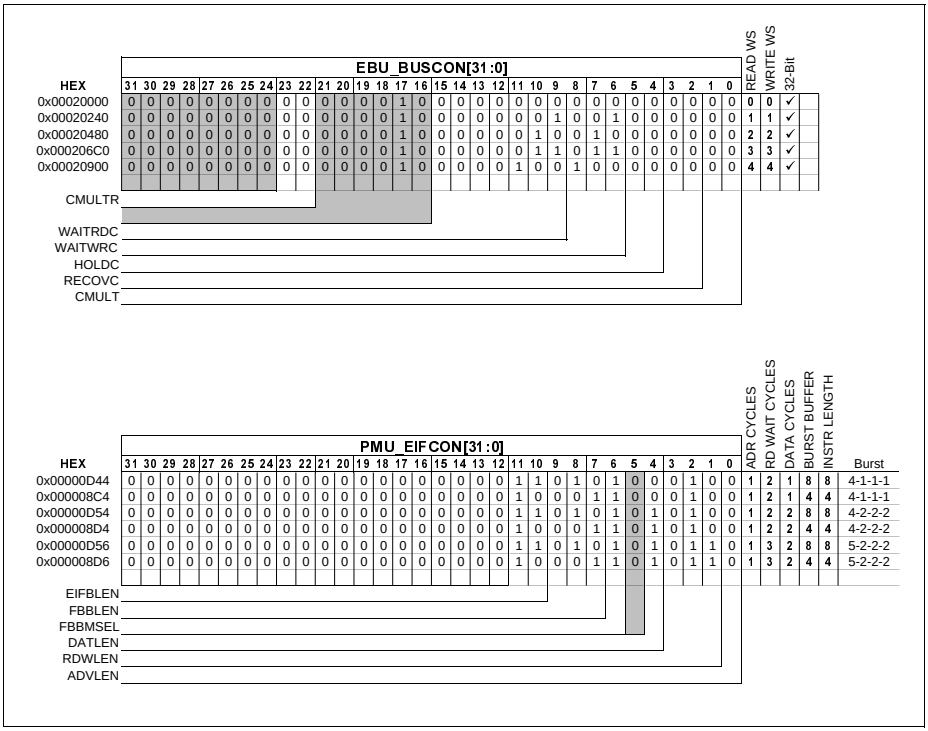

Figure 19 TC1775 EBU/PMU settings

7 References

- **K6R4016V1C**
  4 MBit Static RAM data sheet, Samsung Electronics, 2000

- **M58BW016BT**
  16 MBit Burst Flash memory data sheet, STMicroelectronics, 2001

- **Am29LV160**
  16 MBit Flash memory data sheet, AMD, 1998

- **Am29BL162C**
  16 MBit Burst Flash memory data sheet, AMD, 2000

- **28F160F3**
  16 MBit Burst Flash memory data sheet, INTEL, 2000

- **TC1775 System Units**
  32-Bit Microcontroller, Infineon Technologies, 2001, V2.0 (2001-02)

- **TC1775 Data Sheet**
  32-Bit Microcontroller, Infineon Technologies, 2001, V1.1 (2001-09)

Infineon goes for Business Excellence

“Business excellence means intelligent approaches and clearly defined processes, which are both constantly under review and ultimately lead to good operating results.
Better operating results and business excellence mean less idleness and wastefulness for all of us, more professional success, more accurate information, a better overview and, thereby, less frustration and more satisfaction.”

Dr. Ulrich Schumacher

http://www.infineon.com

Published by Infineon Technologies AG