

# Design and Usage Guidelines for the C2000™ External Memory Interface (EMIF)

TommyLee

## ABSTRACT

The External Memory Interface (EMIF) provides a means for the C2000 device to connect to a variety of external devices, such as interfacing with both flash and SDRAM simultaneously. It was first adopted by the C2000 Microcontroller (MCU) devices starting with the TMS320F2807x and TMS320F2837x product families. This application report provides practical design and usage guidelines for the EMIF.

Supplemental EMIF materials such as throughput benchmarks, register configuration tool, and reference design are discussed in the appendices.

## Contents

1	Introduction .....	3
2	Memory Selection .....	4
3	Hardware Design Considerations .....	6
4	Software Configuration .....	9
5	Usage Optimization .....	10
6	References .....	10
Appendix A	Benchmarks .....	11
Appendix B	EMIF Configuration Tool .....	14
Appendix C	EMIF Reference Design .....	15

## List of Figures

1	Good Four Layer Stackup .....	7
2	Good Six Layer Stackup .....	7
3	"T" Placement .....	8
4	Inline Placement .....	8
5	DMA: Asynchronous Writes .....	11
6	DMA: Asynchronous Reads .....	11
7	DMA: Interleaved Asynchronous Writes and Reads .....	11
8	CPU: Asynchronous Writes .....	11
9	CPU: Asynchronous Reads .....	11
10	CPU: Interleaved Asynchronous Writes and Reads .....	11
11	CLA: Asynchronous Writes .....	12
12	CLA: Asynchronous Reads .....	12
13	CLA: Interleaved Asynchronous Writes and Reads .....	12
14	DMA: Synchronous Writes .....	13
15	DMA: Synchronous Reads .....	13
16	DMA: Interleaved Synchronous Writes and Reads .....	13
17	CPU: Synchronous Writes .....	13
18	CPU: Synchronous Reads .....	13
19	CPU: Interleaved Synchronous Writes and Reads .....	13
20	EMIF Daughtercard Reference Design .....	15

---

### List of Tables

1	Asynchronous Write Sequence .....	3
2	Asynchronous Read Sequence .....	3
3	Common Signal Name Variants .....	6
4	Asynchronous Interface Address Signals .....	6
5	EMIF Signal Substitutes .....	7
6	Common EMIF Fail Behaviors .....	9
7	Multiple EMIF Transactions From Single Access .....	10
8	Asynchronous Configuration .....	11
9	Synchronous Configuration .....	13
10	Guidelines for Evaluation Boards .....	16

### Trademarks

C2000 is a trademark of Texas Instruments.

JEDEC is a trademark of JEDEC Solid State Technology Association.

All other trademarks are the property of their respective owners.

## 1 Introduction

Some C2000 applications require more memory storage than what is available from the internal on-chip memory resources. For these cases, the External Memory Interface (EMIF) controller can be used to access additional off-chip memory devices. The additional external memory can be used for both program instructions and data storage.

### 1.1 Memory Interfaces

The EMIF is compatible with industry standard memory devices which support either asynchronous or synchronous memory interfaces. This section provides details about the supported interface types.

#### 1.1.1 Asynchronous Interface

Asynchronous memory interfaces do not use a common clock signal for signal alignment. Instead, the control signals are interpreted through combinational logic. The EMIF follows a SETUP-STROBE-HOLD sequence that acts upon one memory address and one word per bus operation. The duration for each SETUP, STROBE, and HOLD step is software programmable. For example, the EMIF generates the following SETUP-STROBE-HOLD sequence for asynchronous write and read operations:

**Table 1. Asynchronous Write Sequence**

Signal	Idle	Setup	Strobe	Hold	Idle
$\overline{CS}_{Normal}$	High	Low	Low	Low	High
Address	Invalid	Valid	Valid	Valid	Invalid
Data	Invalid	Valid	Valid	Valid	Invalid
$\overline{WE}$	High	High	Low	High	High
$\overline{CS}_{SS}^{(1)}$	High	High	Low	High	High

<sup>(1)</sup> By default,  $\overline{CS}$  will behave as an ENABLE signal for the memory.  $\overline{CS}$  can also be programmed to behave as a STROBE signal using the Select Strobe mode.

**Table 2. Asynchronous Read Sequence**

Signal	Idle	Setup	Strobe	Hold	Idle
$\overline{CS}_{Normal}$	High	Low	Low	Low	High
Address	Invalid	Valid	Valid	Valid	Invalid
Data	Invalid	Invalid	Valid (at end of step)	Invalid	Invalid
$\overline{OE}$	High	High	Low	High	High
$\overline{CS}_{SS}^{(1)}$	High	High	Low	High	High

<sup>(1)</sup> By default,  $\overline{CS}$  will behave as an ENABLE signal for the memory.  $\overline{CS}$  can also be programmed to behave as a STROBE signal using the Select Strobe mode.

Static RAM (also known as SRAM or ASRAM) and nonvolatile memory (such as NOR flash) are common memory devices which use the asynchronous interface. However, NAND flash is not supported. Non-memory devices, such as FPGAs and digital transceivers, can also make use of the interface to facilitate chip-to-chip data transfer.

### 1.1.2 Synchronous Interface

Synchronous Dynamic RAM (SDRAM) is the only synchronous memory interface supported by the EMIF. Double Data Rate (DDR) and Mobile (LPDDR) SDRAM memories are not compatible. The synchronous interface uses a common clock signal (with fixed timings) to determine when control and data signals are valid. SDRAM commands are defined by JEDEC™ and are automatically encoded by the EMIF once the interface has been configured.

---

**NOTE: F2807x and F2837x do not have full support for SDRAM burst access. Therefore, ASRAM throughput may outperform SDRAM in some situations.**

SDRAM compatibility on F2807x and F2837x is primarily intended to reduce system bill-of-materials (BOM) cost compared to ASRAM. At the time of writing, the retail cost for a 4Mb ASRAM is approximately the same as a 256Mb SDRAM.

---

## 1.2 Memory Access

The contents of the external memory devices are mapped through the EMIF to the C28x program and data memory spaces. All read and write accesses to the EMIF memory space will initiate corresponding read and write operations between the EMIF and external memory.

Each EMIF chip select (CS) signal is assigned to a fixed address range and will automatically assert during read and write operations to its address space. CS signals for asynchronous memories will deassert when idle, while CS signals for SDRAM will typically deassert only when asynchronous memory operations are active. The EMIF will hold CS asserted between pipelined operations.

The assignment of CS signals to memory devices should be considered carefully when designing for the EMIF because CS signals have different:

- **Interfaces:** Each CS space supports only one type of interface (asynchronous or synchronous) -- the interface type can not be changed.
- **Capacity:** The address range of the CS can limit the maximum supported size of the memory.
- **Bus connectivity:** The DMA and CLA may not have access to each CS range.
- **C28x addressability:** The C28x uses a 22-bit address for fetching program instructions. Addresses greater than 0x3FFFFFF can only be reached using the CPU data bus. For guidelines on how to access addresses greater than 0x3FFFFFF with the C28x., see [Accessing External SDRAM on the TMS320F2837x/2807x Microcontrollers Using C/C++](#).

For CS feature details, see the C2000 device-specific data sheets.

## 2 Memory Selection

Memory devices are manufactured in a wide variety of configurations and operating conditions. The process of memory selection can be simplified by first filtering for the memory size and recommended operating conditions.

### 2.1 Memory Size

The scope of potential memory devices can be reduced by only considering those devices which fall within the native address reach of the EMIF. These selected devices will have the capability to provide seamless writes and reads. The maximum capacity supported by the EMIF is determined by the following factors:

#### 2.1.1 CS Address Space

Each CS is assigned a bounded range of 16-bit word addresses that can not be changed. The CS memory reach can be extended virtually by implementing paging with GPIO addressing, but this comes at the cost of additional software overhead.

### 2.1.2 EMIF Address Pins

External data can only be accessed if the EMIF address pins are able to encode the location. Each EMIF module has a finite number of address pins available for encoding. Additionally, some C2000 devices have pin multiplexing conflicts between SDRAM pins and the upper EMIF address pins. In such cases, the SDRAM signals must be given pin multiplexing priority over the address signals when SDRAM is used. Therefore, some address pins may not be available when SDRAM is used.

### 2.1.3 Memory Word Size

The EMIF address pin assignment for asynchronous interfaces will vary depending on the native word size of the memory device (explained later in [Section 3.1](#)). Some EMIF implementations may face pin multiplexing conflicts on the address pins when used with word sizes smaller than 32-bits or when the byte-enable signals are used. Therefore, the memory word size can affect the availability of address pins.

SDRAM address pin assignment does not change with word size.

## 2.2 Operating Conditions

Memory devices have manufacturer specified voltage, frequency, and temperature operating conditions. The supported operating conditions of the memory should meet or exceed those of the host C2000 device in order to avoid memory failures in the system.

### 2.2.1 Voltage

The memory supply voltage is a key characteristic that must be matched with the host device. The DC levels of the memory pins must be compatible with those of the C2000 device. For example, VDDIO is 3.3V nominal on F2807x and F2837x.

The memory should also be able to function within the full VDDIO operating range of the C2000 device if the two devices share a single power supply.

### 2.2.2 Temperature

The supported temperature range is another critical feature because there is no practical way to overcome memory failures when the supported temperature range is exceeded. The selected memory must be able to tolerate the temperature ranges of the target environment.

### 2.2.3 Frequency

The operating frequency of the memory should be fast enough to meet the data throughput requirements of the application. While the theoretical throughput of stand-alone memories can be obtained from manufacturer specifications, the real-world throughput in the system may be reduced by the EMIF hardware and software inefficiencies.

The topics of EMIF optimization ([Section 5](#)) and benchmarks ([Appendix A](#)) are covered later.

### 3 Hardware Design Considerations

#### 3.1 Pin Mapping

The EMIF pins are designed to be connected directly to the equivalently named pins on the memory device. Signal naming conventions vary between manufacturers so the pin descriptions should be confirmed when connecting signals. Some common name variants for memory signals are listed in [Table 3](#).

**Table 3. Common Signal Name Variants**

EMIF Signal	Memory Signal
$\overline{CS}$	$\overline{CE}$
SDCKE	CKE
Dx	DQx, I/Ox,
DQM1	DQMH, UDQM, BHE, UB, BWB, BB
DQM0	DQML, LDQM, BLE, LB, BWA, BA

The exception to the rule of direct pin mapping is that the address pins are shifted for asynchronous memories that have either 8-bit or 16-bit words.

**Table 4. Asynchronous Interface Address Signals**

Memory Signal	32-Bit Memory	16-Bit Memory	8-Bit Memory
A0	EMxA0	EMxBA1	EMxBA0
A1	EMxA1	EMxA0	EMxBA1
A2	EMxA2	EMxA1	EMxA0
A3	EMxA3	EMxA2	EMxA1
A4	EMxA4	EMxA3	EMxA2
A5	EMxA5	EMxA4	EMxA3
...	...	...	...
A <sub>y</sub>	EMxA <sub>y</sub>	EMxA <sub>y-1</sub>	EMxA <sub>y-2</sub>

This shifting scheme is designed to maintain the EMIF memory capacity for smaller word sizes by supplementing the number of available address pins with the otherwise unused EMxBAy pins. However, some EMIF implementations may lose address reach because of pin multiplexing conflicts.

For example, EMIF1 CS2 on F2837x can support up to 4MB of memory by using EM1A0 through EM1A19 for a 32-bit memory (1M x 32b) with only 32-bit word access. In order to maintain 4MB of capacity with a 16-bit memory (2M x 16b), EM1BA1 is used to supplement EM1A0 through EM1A19. However, EM1BA1 has a pin multiplexing conflict with EM1A19, so the overall capacity on CS2 is reduced to 2MB (1M x 16b) when a 16-bit memory is used.

### 3.2 Pin Multiplexing Conflicts

Pin multiplexing conflicts may arise when pin availability is strained or when mixed memory types are used. In these situations, some EMIF signals may be substituted or omitted with limitations. Potential substitutions are described in [Table 5](#).

**Table 5. EMIF Signal Substitutes**

EMIF Signal	Description
DQMx	Dynamic control of the byte enable signals is not required for 16-bit memories because the C28x is 16-bit word addressable. The memory byte enable signals can be tied low (permanently enabled) for 16-bit memories. Byte enable signals for 32-bit memories can also be tied low if 16-bit writes and reads are not used.
$\overline{WE}$ , $\overline{OE}$	Asynchronous memories that support $\overline{CS}$ controlled operations can enable the Select Strobe mode and use RNW to directly control $\overline{WE}$ and externally invert RNW to control $\overline{OE}$ .
CKE	The CKE signal can be tied high (permanently enabled) if the SDRAM Self Refresh, Power Down, and Clock Suspend modes are not used.
Ax	Upper address signals for asynchronous memories can be controlled by GPIO instead of EMIF. The application software will need to treat the GPIO-controlled address spaces as virtual pages. SDRAM signals must be given pin multiplexing priority over Ax signals when SDRAM is used.

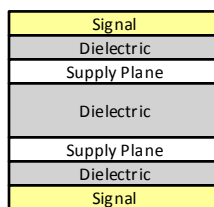
### 3.3 Layout

The EMIF is typically the interface that generates the most pin activity and the highest switching rates on the board. Attention should be given to the proper routing and placement of memory devices. An in-depth review of layout theory and techniques is beyond the scope of this document, but the following guidelines may be useful.

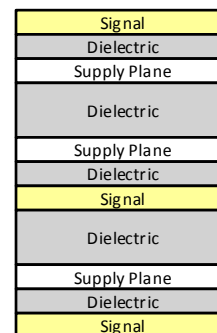
#### 3.3.1 Routing

Propagation delay and signal skew are both crucial parameters when routing for a high-speed parallel bus. An excess of either parameter can lead to significant performance degradation or memory failures if setup and hold timings are not satisfied. A timing budget should be created to analyze how much propagation delay and skew the system can tolerate for the target operating frequency.

Cross-talk is another common concern for EMIF designs because the signals are often constrained within a small area. The potential for cross-talk in congested areas can be controlled by routing signals directly over continuous power planes between the MCU and memory devices. Continuous planes will also help to distribute power readily and evenly to the memory supply pins during peak activity. [Figure 1](#) shows examples of board layer stackups that promote good signal integrity.



**Figure 1. Good Four Layer Stackup**



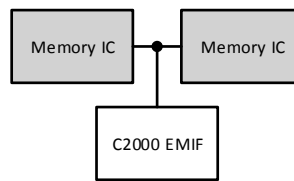
**Figure 2. Good Six Layer Stackup**

### 3.3.2 Multiple Memories

EMIF modules with multiple CS signals will have pin buffers that are strong enough to drive multiple memory devices connected in parallel. It is generally acceptable to connect as many memory devices as there are CS signals. For example, an EMIF module with four CS signals will be able to drive the load of four memory devices in parallel.

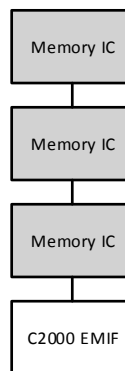
However, multiple memory devices may increase the occurrence of unwanted signal reflections. Distortions can be minimized by optimizing the memory placement. For example, two identical memories can be routed to the EMIF using a "T" pattern as shown in [Figure 3](#).

Ideally, the memory devices are placed close to each other and the memory device endpoints are equidistant from the common signal branch. Short traces after the branch will produce more manageable reflections than long traces. Equal trace lengths between the memory devices help to balance signal propagation, which is important if the memory devices are accessed simultaneously (for example, if emulating a single 32-bit memory device by using two 16-bit memory devices in parallel).



**Figure 3. "T" Placement**

Another option for placing memory devices is with an inline pattern as shown in [Figure 4](#). This topology may be easier to implement for systems that use more than two memory devices or use different memory types. The goal is to mimic continuous signal traces by using short stub lengths when branching off to inline memory devices. The intermediate memory devices will produce minimal reflections when the EMIF is driving control and address signals down the bus. However, termination may be required at the endpoint memory devices to reduce reflections when the intermediate memory devices are driving data onto the bus in response to read operations.



**Figure 4. Inline Placement**



## 4 Software Configuration

Each CS has an independent set of registers that must be configured before accessing the attached memory. These configuration values must match the memory topology and satisfy the timing requirements in the memory and C2000 device-specific data sheets.

### 4.1 Memory Timings

CS timings are expressed as multiples of the EMIF peripheral clock cycles. For example, suppose that an asynchronous memory on CS2 has a 12 ns write strobe requirement. The 12 ns write strobe duration can be satisfied by asserting the  $\overline{WE}$  and data signals for three EMIF peripheral clock cycles at 200 MHz (5 ns period): 3 cycles x 5 ns = 15 ns.

The W\_STROBE register field is defined as the number of required cycles minus one, so in software, the value of two would be written to the field: W\_STROBE = 3 cycles - 1 = 2.

Registers settings for SDRAM on CS0 follow the same convention with the added consideration that the SDRAM EMxCLK signal is derived directly from the EMIF peripheral clock. Therefore, the EMIF peripheral clock speed should be configured to stay within the recommended operating frequencies for both the SDRAM and EMIF module.

The process of calculating configuration register values can be tedious and error-prone. A spreadsheet tool is available for generating and checking EMIF configuration values. For additional information, see [Appendix B](#).

### 4.2 Configuration Debug

Memory failures are common during development because of the large number of signals used and high switching speeds involved. Common issues can be diagnosed using a methodical approach that categorizes fails as being caused by timing sensitivities or by continuity problems.

#### 4.2.1 Timing Sensitivities

Timing failures often appear to be random and can usually be eliminated by setting all timing related fields (except for SDRAM Refresh Rate) to their maximum supported values so that memory state transitions are slowed down. Further delay can be introduced by dividing down the EMIF peripheral clock and device SYSCLK as well. If the memory behavior improves with slower settings, the sensitive timing parameters can be identified through the process of elimination.

This approach of identifying timing violations can also be adapted to test for timing margin between the EMIF and external memory devices. For example, consider a system that will deploy with the EMIF running at 200 MHz and with W\_STROBE configured to use three cycles (3 cycles x 5 ns = 15 ns). For stress testing purposes, W\_STROBE can be reduced to use only two cycles (2 cycles x 5 ns = 10 ns). Normal write operations with 10 ns of strobe would imply that the system has at least 5 ns of strobe timing margin. Bad write operations would imply that the system has less than 5 ns of strobe timing margin.

#### 4.2.2 Continuity Problems

Continuity problems usually exhibit repeatable fail patterns when signals are shorted. Resistive or floating signals may appear to abruptly shift between working and non-working states when the pin input randomly changes between high and low levels. Common continuity fail behaviors are described in [Table 6](#).

**Table 6. Common EMIF Fail Behaviors**

EMIF Pin	Fail Behavior
CS, CLK, SDCKE, $\overline{OE}$ , RAS, CAS	Reads return a repeating static value for multiple addresses
Ax, BAx	Data from one address appears to be mirrored at another address
Dx	Data bits are stuck high or low for multiple addresses
DQMx	Data bytes are stuck high or low for multiple addresses
$\overline{WE}$	Writes appear to have no effect

## 5 Usage Optimization

The EMIF performance is influenced by a number of hardware and software factors. Apparent sources of performance limitations (such as operating speed and word size) are related to the choice of memory device. These behaviors are specified in the manufacturer data sheets and will not be covered in this section.

Less obvious factors (such as memory access size and bus efficiency) are explained in the following sections. Memory throughput may suffer when software use conditions are not optimized for the C2000 EMIF design implementation.

### 5.1 Access Size

The CPU, DMA, and CLA are each capable of using 16-bit and 32-bit access modes for data operations. Each access mode takes the same number of cycles to execute, which gives the 32-bit access mode an advantage over the 16-bit access mode in terms of raw throughput.

Further, the EMIF is connected to the CPU, DMA, and CLA using a 32-bit data bus that can be fully utilized for all external memory word sizes.

Internal data bus transactions are pipelined into consecutive memory transactions by the EMIF if the external word size is smaller than 32-bits. In such instances, the external EMIF operations are optimized: the CS signal for asynchronous memory may remain asserted between words and SDRAM may use burst mode access. Therefore, higher throughput can be achieved by accessing the EMIF memory using 32-bit reads and writes as opposed to 16-bit reads and writes.

Examples of write transactions for various word sizes are shown in [Table 7](#).

**Table 7. Multiple EMIF Transactions From Single Access**

Internal Write Value	External Memory Word Size	Write 1	Write 2	Write 3	Write 4
0x12345678	32-bits	0x12345678	-	-	-
	16-bits	0x5678	0x1234	-	-
	8-bits	0x78	0x56	0x34	0x12
0x1234	32-bits	0x00001234	-	-	-
	16-bits	0x1234	-	-	-
	8-bits	0x34	0x12	-	-

### 5.2 Bus Latency

Internal bus access to the EMIF from CPU, DMA, and CLA must pass through a synchronization bridge because the endpoints can be configured to operate at different clock frequencies.

Each random access to the EMIF memory space will experience three SYSCLK cycles of synchronization latency. This latency can be reduced to two SYSCLK cycles per operation if the synchronization bridge is fully loaded by a continuous stream of pipelined accesses. Throughput is thus maximized by using block memory transfers rather than single word transfers.

The DMA is able to achieve high bus efficiency with very little processor overhead. Background data transfers using 32-bit DMA access are recommended whenever possible. Optimized assembly instructions on the CPU and CLA pipelines can outperform DMA bus efficiency under some conditions, but at the cost of significant processor overhead.

## 6 References

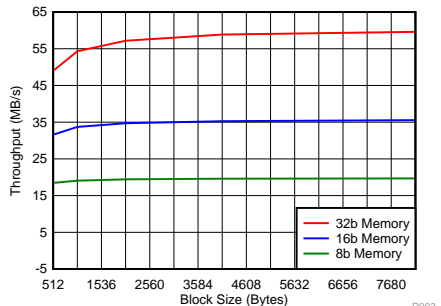
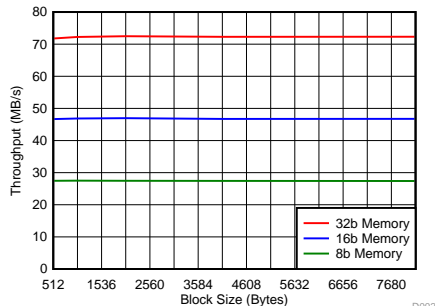
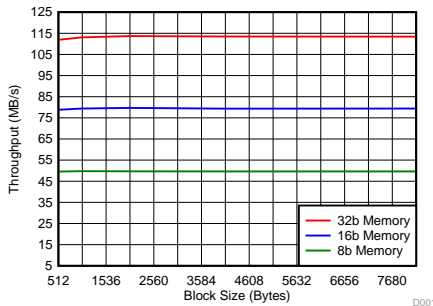
[Accessing External SDRAM on the TMS320F2837x/2807x Microcontrollers Using C/C++](#)

# Benchmarks

## A.1 Asynchronous Throughput

**Table 8. Asynchronous Configuration**

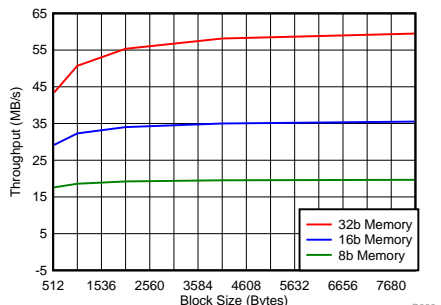
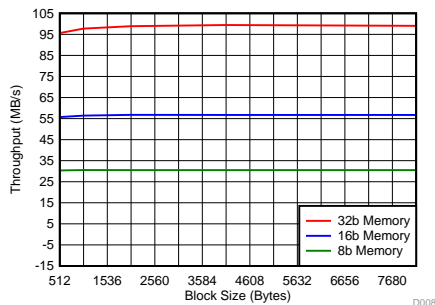
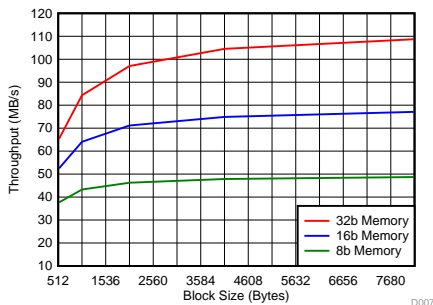
Parameter	Value
Device	F2837x
SYSCLK	200 MHz
EMIF CLK	SYSCLK / 1
W_SETUP - W_STROBE - W_HOLD	0 - 0 - 0
R_SETUP - R_STROBE - R_HOLD	0 - 3 - 0
TA	0
Memory Word Size (External Data Bus Size)	32b, 16b, 8b
Internal Bus Access Size	32b



**Figure 5. DMA: Asynchronous Writes**

**Figure 6. DMA: Asynchronous Reads**

**Figure 7. DMA: Interleaved Asynchronous Writes and Reads**

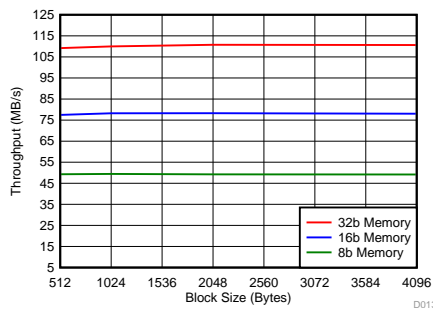


**Figure 8. CPU: Asynchronous Writes**

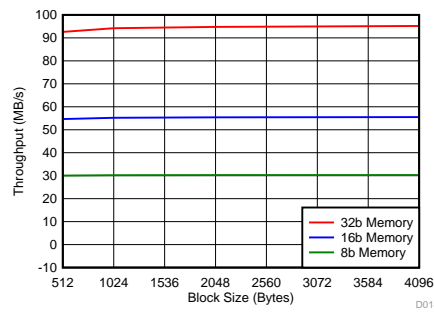
**Figure 9. CPU: Asynchronous Reads**

**Figure 10. CPU: Interleaved Asynchronous Writes and Reads**

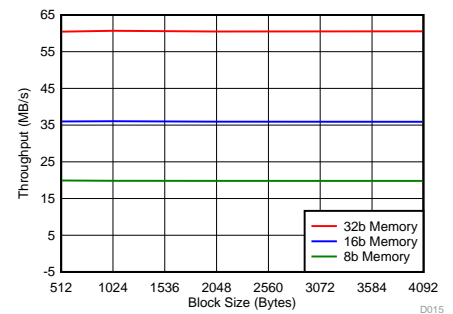
**NOTE:** CPU transfers performed using assembly routine memcpy\_fast\_far.asm distributed through [C2000Ware](#). Directory: ~\C2000Ware\libraries\dsp\FPU\c28\source\utility.



**Figure 11. CLA: Asynchronous Writes**



**Figure 12. CLA: Asynchronous Reads**



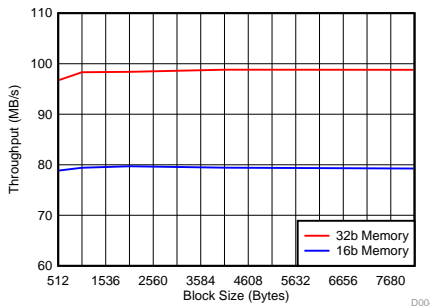
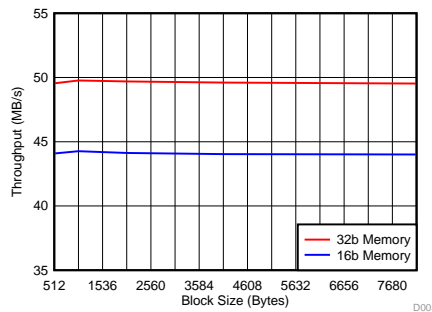
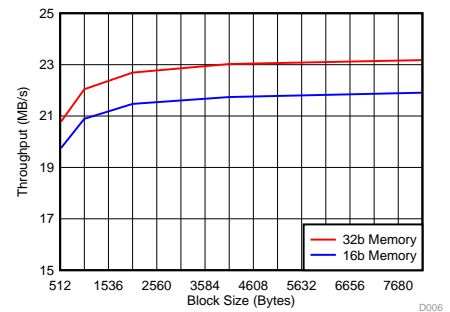
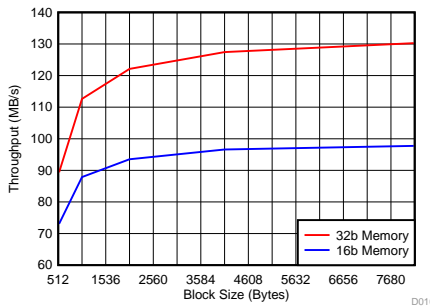
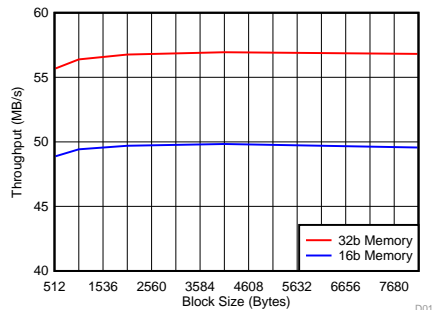
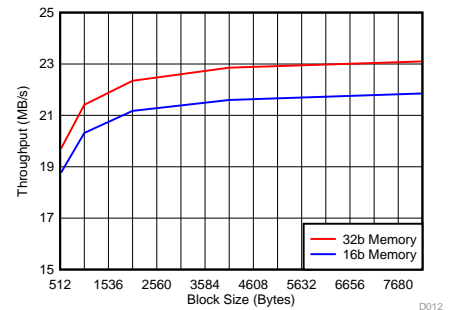
**Figure 13. CLA: Interleaved Asynchronous Writes and Reads**

**NOTE:** CLA transfers performed using assembly routine from EMIF\_DC examples distributed through [C2000Ware](#). Directory: ~\C2000Ware\device\_support\12837xd\examples\cpu1\emif\_dc\_cla.

## A.2 Synchronous Throughput

**Table 9. Synchronous Configuration**

Parameter	Value
Device	F2837x
SYSCLK	200 MHz
EMIF CLK	SYSCLK / 2
Memory Word Size (External Data Bus Size)	32b, 16b
Internal Bus Access Size	32b


**Figure 14. DMA: Synchronous Writes**

**Figure 15. DMA: Synchronous Reads**

**Figure 16. DMA: Interleaved Synchronous Writes and Reads**

**Figure 17. CPU: Synchronous Writes**

**Figure 18. CPU: Synchronous Reads**

**Figure 19. CPU: Interleaved Synchronous Writes and Reads**

**NOTE:** CPU transfers performed using assembly routine memcopy\_fast\_far.asm distributed through [C2000Ware](#). Directory: ~\C2000Ware\libraries\dsp\FPU\c28\source\utility.

## **EMIF Configuration Tool**

---

---

---

### **B.1 Overview**

A spreadsheet tool for generating and checking EMIF configuration register values is distributed with [C2000Ware](#) (location: ~\C2000Ware\boards\TIDesigns\F28379D\_EMIF\_DC\C2000-EMIF\_ConfigurationTool.xlsx).

The register configuration sheet can calculate new EMIF register values by using specification values from the memory and C2000 device-specific data sheets. The register decode sheet helps to confirm and compare existing register values. Cells shaded in green are user-writeable.

The Analysis ToolPak Add-In must be installed and enabled in order to use the tool.

### **B.2 Configuration Sheet**

The configuration sheet combines characteristics from both the C2000 device and external memory devices to calculate CS configuration values. Calculated values are shown at both the register and field levels for ease of use.

The sheet is comprised of two sections. The first section represents characteristics of the C2000 EMIF and PCB. Implementation-dependent values like EMxCLK and PCB Delay are determined by the user. The EMIF timing values are device-dependent and should be taken from the C2000 device-specific data sheet.

The second section focuses on the characteristics of the memory devices that are used with the EMIF. These values are taken from the memory data sheet. Each value should be readily available from the manufacturer, but the terminology may vary. In cases where the parameter name cannot be matched, use the behavioral description to find the equivalent parameter.

### **B.3 Decode Sheet**

The decode sheet converts full register values into decimal numbers as defined by the register field boundaries. Timing related fields will also use the given the EMIF clock frequency to calculate the effective duration for each parameter. The calculated times can be compared against the manufacturer's memory data sheets to confirm that timing requirements are satisfied.

A secondary function of the decode sheet is to provide a means of visualizing differences between two sets of register configurations. This can be useful for comparing register settings that were calculated using different assumptions or for debugging systems that have different configuration values.

## EMIF Reference Design

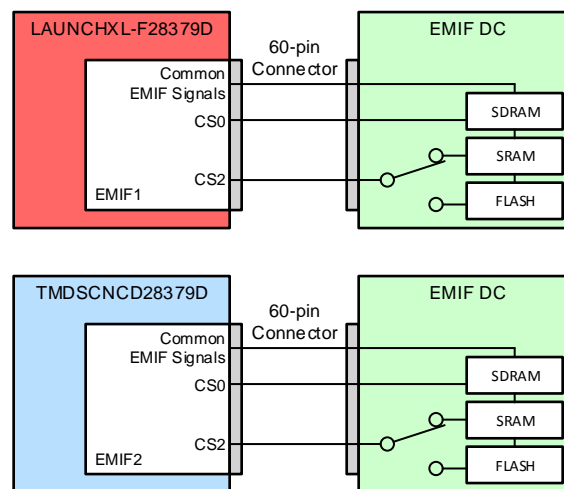
### C.1 Overview

An EMIF hardware reference design is distributed with [C2000Ware](#) (directory: ~\C2000Ware\boards\TIDesigns\F28379D\_EMIF\_DC).

Usage examples for the reference design are included with the f2837xd software examples and are denoted by the "emif\_dc\_" prefix.

The reference design places three 16-bit memory devices (SDRAM, SRAM and FLASH) on a daughtercard that can dock with the high density 60-pin connector on the LAUNCHXL-F28379D and TMDSCNCD28379D evaluation boards.

A number of unique design implementations were driven by board compatibility concerns and the limited availability of EMIF signals across the 60-pin connector. These features are explained in the following sections.



**Figure 20. EMIF Daughtercard Reference Design**

### C.2 Chip Select Signals

Only two chip select signals are routed to the 60-pin connector: CS0 and CS2. In order to support two asynchronous memories with a single CS signal, a GPIO-controlled analog switch is used to change the destination of the CS2 signal between SRAM and FLASH. This configuration requires software to arbitrate between SRAM and FLASH access, and it forces the SRAM and FLASH to share a single memory space.

Custom boards should use direct CS<sub>n</sub> routing wherever possible in order to avoid similar usage complexities.

The original design of the daughtercard used an analog switch (TS5A3159DBVR) that distorted CS2 signal transitions because of high parasitic capacitance. A pin-compatible replacement switch (TS5A3157DBVR) was found to have much less capacitance and associated distortion.

### C.3 Address Signals

Address signals above A12 are not routed to the 60-pin connector. External memory beyond this reach can only be accessed by controlling the higher memory address pins with software-controlled GPIO signals. External memory reach is extended with this implementation, but software takes on the additional responsibilities of monitoring and changing virtual pages as required.

Custom boards should use direct address mapping for seamless memory access. GPIO signals can be used to increase external memory reach if required.

### C.4 Hardware Differences Between Evaluation Boards

There are number of hardware differences between the evaluation boards that require special handing. A summary of differences and guidelines are listed in [Table 10](#) and explained in the following sections.

**Table 10. Guidelines for Evaluation Boards**

Difference	LAUNCHXL-F28379D R1.x	LAUNCHXL-F28379D R2.0+	TMDSCNCD28379D
<a href="#">EMIF used for 60-pin connector</a>	EMIF1	EMIF1	EMIF2
<a href="#">Reference clock</a>	10 MHz	10 MHz	20 MHz
<a href="#">_LAUNCHXL_F28379D</a>	Add to Predefined Symbols	Add to Predefined Symbols	Do not define
<a href="#">EMxSDCKE available on 60-pin connector</a>	No (control with GPIO)	Yes	Yes
<a href="#">EMxA12 available on 60-pin connector</a>	No	Yes	No
<a href="#">Daughtercard J2 header for EMxA12</a>	Jump EM1A12 from LaunchPad BoosterPack header J5-48 (or P52) to J2-2 (EM_A12)	Place jumper to connect J2-1 (GPIO_EM_A12) and J2-2 (EM_A12)	Place jumper to connect either: <ul style="list-style-type: none"> <li>J2-1 (GPIO_EM_A12) and J2-2 (EM_A12) for GPIO-controlled low, or</li> <li>J2-2 (EM_A12) and J2-3 (GND) for static low</li> </ul>
<a href="#">0Ω resistors required for 60-pin connector</a>	No	Yes	No

#### C.4.1 EMIF Assignment

LAUNCHXL-F28379D uses EMIF1 signals while TMDSCNCD28379D uses EMIF2 signals for their respective 60-pin connectors. The emif\_dc software examples use the "\_LAUNCHXL\_F28379D" predefined symbol to determine which EMIF should be used with the daughtercard. EMIF1 is selected when the symbol is defined, otherwise EMIF2 is selected.

#### C.4.2 Clock Frequency

LAUNCHXL-F28379D uses a 10 MHz reference while TMDSCNCD28379D uses a 20 MHz reference. By default, the C2000Ware examples expect a 20 MHz input clock. The examples are directed to use a 10 MHz input clock when the predefined symbol "\_LAUNCHXL\_F28379D" is declared.

#### C.4.3 SDRAM Clock Enable

LAUNCHXL-F28379D R1.x does not have the correct EMxSDCKE signal routed to the 60-pin connector. In order to use SDRAM with LAUNCHXL-F28379D R1.x, the clock enable signal must be driven high by GPIO. The Power Down, Clock Suspend, and Self Refresh modes will not be available without the native EMxSDCKE signal.

#### C.4.4 EMIF A12 Signal

Header J2 on the daughtercard provides flexibility in handling the EMIF A12 signal.



**C.4.4.1 LAUNCHXL-F28379D R2.0+**

LAUNCHXL-F28379D R2.0 and later should install a jumper to use the native EM1A12 signal coming from the 60-pin connector.

**C.4.4.2 LAUNCHXL-F28379D R1.x**

LAUNCHXL-F28379D R1.x does not route EM1A12 to the 60-pin connector, however the signal is available from the BoosterPack headers. Full EM1A12 functionality can be realized by connecting the signal from BoosterPack header J5 to the daughtercard header J2 with a jumper wire.

**C.4.4.3 TMDSCNCD28379D**

TMDSCNCD28379D does not have a native A12 signal on EMIF2 so it must be held low on the daughtercard for proper memory operation. This can be done by driving static low with a GPIO pin or by connecting the signal to GND.

Dynamic control of A12 with GPIO is possible for asynchronous memories, but some SDRAMs may malfunction when A12 is asynchronously held high.

**C.4.5 Signal Isolation on LAUNCHXL-F28379D R2.0 +**

LAUNCHXL-F28379D R2.0 and later include 0Ω resistor footprints to isolate certain EMIF signals from the 60-pin connector and BoosterPack headers.

By default, the signals are connected to the BoosterPack headers and are isolated them from the 60-pin connector. The EMIF signals must be connected to the 60-pin header before using the EMIF daughtercard. Isolating the signals from the BoosterPack is recommended.

Consult the LAUNCHXL-F28379D schematic to determine which signals are isolated.

---

## Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

<b>Changes from Original (October 2017) to A Revision</b>	<b>Page</b>
• Update was made in the Abstract. ....	1

---

## IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ("TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products (<http://www.ti.com/sc/docs/stdterms.htm>), [evaluation modules](#), and [samples](http://www.ti.com/sc/docs/sampterm.htm) (<http://www.ti.com/sc/docs/sampterm.htm>).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2018, Texas Instruments Incorporated