OMAP2420/TUSB6010B DMA Configuration

Wes Ray

ABSTRACT

To achieve the highest USB throughput using the TUSB6010B USB 2.0 high-speed dual role USB controller, the host processor (in this case the OMAP 2420) must use its direct memory access (DMA) controller to exchange data with the TUSB6010B over the NOR-FLASH-like interface. The OMAP2420 memory controller, direct memory access controller, and TUSB6010B must be properly configured to achieve the best possible throughput. This application report describes the parameters that must be properly configured and provides software pseudo-code for implementing the setup.

Note: The 2420 configuration detailed in this application report is intended to be used on the ES2.1 or later version of 2420 silicon. Using this configuration on silicon versions prior to ES2.1 is not ensured to work as described.

1 Introduction

1.1 Overview

The TUSB6010B is a USB 2.0 high-speed dual role USB controller designed for seamless interface to an external host processor through the NOR-FLASH-like interface. The NOR-FLASH-like interface is a multiplexed, 16-bit interface with support for burst and single read/write access. It can operate in both synchronous and asynchronous modes. Data can be exchanged with the host in either DMA or non- DMA mode. When using the synchronous chip select, data is transferred in either 8 × 16 or 16 × 16 burst modes.

The purpose of this application report is to provide an example of how to interface the TUSB6010B (hereafter referred to as the 6010) with the OMAP2420 host processor (hereafter referred to as the 2420), using the OMAP2420 DMA controller to synchronously burst data over the NOR interface. There are three areas of the 2420/6010 system that must be properly configured to ensure the highest throughput on the USB; they are the: 2420 GPMC, 2420 DMA controller, and 6010 configuration registers. Figure 1 shows how the 2420 and 6010 are connected in a USB application.

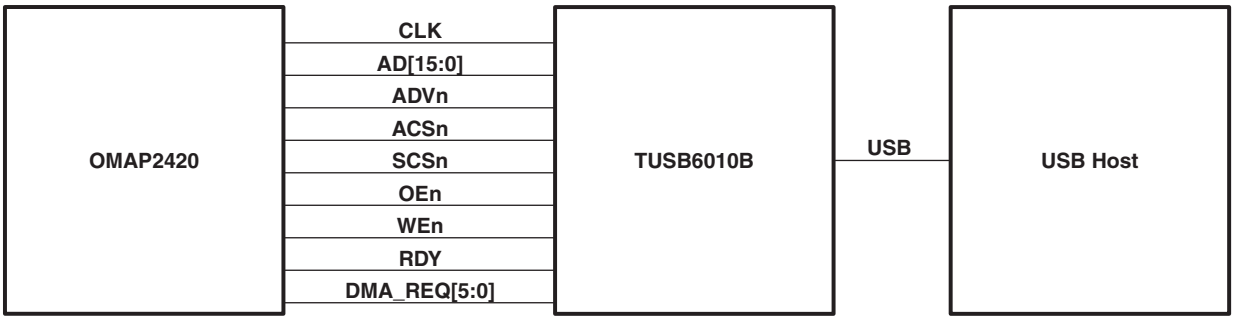

Figure 1. 2420/6010 System Block Diagram

1.2 Data Movement with DMA

The most efficient way to move data between the 2420 and the 6010 FIFO(s) is to use the 2420 DMA controller. The DMA controller can be configured to burst data in 8 × 16 (16 bytes) or 16 × 16 (32 bytes) blocks. The DMA controller can transfer data to the 6010 FIFO(s) using both the synchronous and asynchronous chip selects. However, for optimum throughput, the synchronous chip select should be used.

The 6010 has six DMA request lines that are connected to the 2420 and are used to signal the DMA controller to begin the transfer of data to/from the 6010. DMA request lines 0 and 1 are available on dedicated pins and GPIO 0 and 1 respectively. DMA request lines 2–5 are available on GPIO 2–5 respectively. If the DMA requests on the GPIO pins are used, the 6010 GPIO configuration register must be properly programmed, see chapter 4. Proper multiplexing of the 2420 SYS.nDMAREQn lines is also required. See the 2420 technical reference manual (TRM) (literature number SWPU064) for more information.

1.3 Assumptions

It is assumed that the reader has a general knowledge of how to program the 2420 GPMC and DMA controller or can at least find the proper sections in the 2420 TRM. Knowledge in these areas is not essential but will aide in a timely implementation of concepts presented in this application report. It is recommended that the reader take the time to read the DMA and memory subsystem chapters of the 2420 TRM. A working knowledge of the 6010 is also assumed. It is recommended that the reader first read the datasheet, data manual, and MUSBMHDRC programmer's guide before attempting to implement the configuration described in this application report.

2 2420 GPMC Configuration

The GPMC is configured using seven configuration registers GPMC_CONFIG(1-7)_N, see the memory subsystem chapter of the 2420 TRM. Once the GPMC has been configured, only the ATTACHED_DEVICE_PAGELENGTH parameter in GPMC_CONFIG1_N needs to be changed to switch between 8 × 16 and 16 × 16 burst modes. For maximum throughput, 16 × 16 burst mode should be used. The following configuration values can be used when the 2420 is in clock configuration 2:

16×16 Synchronous Chip Select Values, Assuming 110 MHz 2420 L3 Clock

GPMC_CONFIG1_N = 0x7B421201 (1)

GPMC_CONFIG2_N = 0x000C0C00

GPMC_CONFIG3_N = 0x00020200

GPMC_CONFIG4_N = 0x0B040B04

GPMC_CONFIG5_N = 0x02090C0D

GPMC_CONFIG6_N = 0x000002C2

GPMC_CONFIG7_N = 0x00000F7F (base address 0x3F000000)

(1) The device wait count register in the 6010 must be set to 1.

When running the 2420 in clock configuration 1, these values must be modified to align with the timing in the 6010 datasheet.

For both 6010 ES1.0 and ES2.0 silicon, the GPMC_FCLK_DIVIDER must be set to 1 in GPMC_CONFIG1_N[1:0]. This will make the external GPMC.CLK equal to half of the internal GPMC.FCLK, which is the same frequency as the 2420 L3 clock. The L3 clock should be double the desired GPMC.CLK which is connected to the 6010 CLK input pin. The 2420 TRM power, reset, and clock management chapter shows the various clocking schemes and describes their configuration. See the 6010 Datasheet for NOR Clock frequency limits and timing. For 6010 ES2.0 silicon, WAIT WRITE MONITORING should be disabled in GPMC_CONFIG1_N [21].

See the 2420 TRM for information related to the WAIT pin selection and base address as these are system independent settings.

- WAIT_PIN_SELECT – GPMC_CONFIG1_N[17:16]
- MASK_ADDRESS/ BASE_ADDRESS – GPMC_CONFIG7_N[11:8]/[5:0]

3 2420 DMA Configuration

3.1 Overview

The 2420 system DMA controller must be configured to use the hardware-synchronized transfer type. Once a logical DMA channel is configured and enabled, the channel becomes active and transfers start when the DMA request line is asserted from the 6010.

3.2 Block Size and Addressing

A DMA transfer block consists of a number of frames. Each frame consists of a number of elements and each element can have a size of 8, 16, or 32 bits. The number of frames, number of elements per frame, and element size are common for both the source and destination. However, the way in which the data is represented (addressing profile/mode) is independently programmable for the source and destination devices. For the 2420 system memory it is assumed that the addressing will be postincrement. The transfer block size is calculated as follows:

transfer block size = (number of frames) $\times$ (number of elements per frame) $\times$ (element size)

An example (see pseudo-code below), of how to transfer 15 512 byte USB packets using a 16 $\times$ 16 burst on the NOR interface is given:

transfer block size = 15 packets $\times$ 512 bytes per packet = 7680 bytes

6010 NOR burst port size is 16 $\times$ 16 $\geq$ 32 bytes or 8 $\times$32-bit burst size (see DMA4_CSDPn[SRC/DST_BURST_EN])

32 bytes burst yields 8 elements of 4 bytes (quadlets each)

number of frames = 7680 / 8 / 4 = 240

Thus, it will take 240 16 $\times$ 16 bursts on the NOR interface to transfer 15 512 byte USB packets between the 2420 and 6010.

The addressing for the 6010 FIFO is not obvious. When using the synchronous chip select, the 6010 FIFO is accessed 32 bits at a time at only the FIFO base address. In order to achieve a burst to this type of addressing scheme, double-index increment addressing must be used on the 6010 FIFO. However the frame index must be equal to the negative number of the burst size of the 6010 FIFO port to ensure that at the end of each burst, the DMA address will decrement back to the start of the 6010 FIFO. See pseudo code below and 2420 TRM DMA addressing modes section for details.

3.3 DMA Request

There are six external DMA request lines on the 2420, SYS.nDMAREQn. See the 2420 TRM for information related to the multiplexing of these pins as these are system dependent settings. Mapping of these DMA request lines for the SYNCHRONIZATION_CONTROL field of the DMA4_CCRn register can be found in the DMA chapter of the 2420 TRM.

3.4 Configuration

After a software or hardware reset, all logical channel registers should be programmed to their default values since most fields in these registers are undefined at reset, see sample code. The priority arbitration rate and maximum 2420 DMA FIFO depth must be configured via the DMA4_GCR prior to any DMA logical channel configuration. If interrupts are to be used, (DMA interrupt setup is beyond the scope of this document) they must be enabled in the DMA4_IRQENABLE_L0–DMA4_IRQENABLE_L3 and DMA4_CICR registers. A 16 × 16 hardware-synchronized DMA logical channel can be programmed as follows.

1. DMA4_GCR:
   - Arbitration rate. This field is system dependent. It is left to the user to find the optimum setting.
   - Maximum FIFO channel depth. Should be set to 32.
2. DMA4_CSDP:
   - Transfer element size (8-bit, 16-bit or 32-bit), DMA register bits DMA4_CSDP[1:0]. Set to 0x10 for 32 bits scalar.
   - Read and write port access types (single/burst), DMA register bits DMA4_CSDP[8:7] and DMA4_CSDP[15:14]. For a 6010 read channel, DMA4_CSDP[8:7] should be set to 0x10 to allow for fast access to the 2420 system RAM and DMA4_CSDP[15:14] should be set to 0x10 to allow for a 16 × 16 burst of the 6010 FIFO. For a 6010 write channel, DMA4_CSDP[8:7] should be set to 0x10 to allow for fast access to the 2420 system RAM and DMA4_CSDP[15:14] should be set to 0x10 to allow for a 16 × 16 burst of the 6010 FIFO.
   - Source and destination endianness, DMA register bits DMA4_CSDP[21:20] and DMA4_CSDP[19:18]. The 6010 should always be set for little endian, 0x00. Set this field accordingly for the 2420 system RAM, 0x00 or 0x01.
   - Write mode (posted or non-posted), DMA register bits DMA4_CSDP[17:16]. All read/write transactions should be mapped on the write command as posted except for the last transaction in the transfer (mapped on a write none posted). This is a setting of 0x10.
   - Source or destination packed or non-packed (if the element size is less than the read/write port size), DMA register bits DMA4_CSDP[6] and DMA4_CSDP[13]. Packing should not be used for read or write channels, always set these bits to 0.
3. DMA4_CEN:
   Number of elements (quadlets) per frame, 4 for an 8 × 16 NOR burst or 8 for a 16×16 NOR burst.
4. DMA4_CFN:
   Number of frames per transfer block, must be rounded up number of the USB transfer size (see pseudo-code below).
5. DMA4_CSSA and DMA4_CDSA:
   Source and destination start address (aligned with transfer element size). Both the source and destination addresses must be aligned on a 32-byte burst boundary. Use the 2420 system memory base buffer address (post-increment mode) and the 6010 FIFO base address (double-index mode).
6. DMA4_CCR:
   - Read and write port addressing modes, DMA register bits DMA4_CCR[13:12] and DMA4_CCR[15:14]. For a 6010 read channel, DMA4_CCR[13:12] should be set to 0x11 and DMA4_CCR[15:14] should be set to 0x01. For a 6010 write channel, DMA4_CCR[13:12] should be set to 0x01 and DMA4_CCR[15:14] should be set to 0x11.
   - Prefetch enabled or disabled, DMA register bit DMA4_CCR[23]. For a 6010 read channel, DMA4_CCR[23] should be set to 0. For 6010 writes channels set this bit to 1.
   - Low or high priority, DMA register bit DMA4_CCR[6]. Always set this bit to 1.
   - DMA request number (corresponding to the incoming DMA request line to synchronize the transfer to), DMA register bits DMA4_CCR[4:0] and DMA4_CCR[20:19]. Set appropriate DMA request number to be used for this channel. See DMA4_CCR description in the TRM for mapping tables.
   - Amount of data to transfer per DMA request (via frame and block synchronization bits), DMA register bits DMA4_CCR[5] and DMA4_CCR[18]. For both read and write channels, Set DMA4_CCR[5] to 1 and DMA4_CCR[18] to 0 to indicate that an entire frame is to be transmitted when a DMA request is received.
   - Whether the sender of the DMA requests is the source or the destination, DMA register bits DMA4_CCR[24]. For a 6010 read channel, set to 1, source initiated transfer. For a 6010 write channel, set to 0, destination initiated transfer.
7. DMA4_CSEI, DMA4_CSFI, DMA4_CDEI and DMA4_CDFI:

Source and destination element and frame indexes (depending on addressing mode)

8. DMA4_CSFI or DMA4_CDFI:
   Number of elements per packet (only for synchronization to a packet, and only one of these registers depending on whether the source or destination is sending the DMA requests)
9. Enable the transfer via the enable bit in the channel DMA4_CCR register, DMA register bit DMA4_CCR[7].

Once the above configuration has been set for a logical channel, the first transfer will start when the associated DMA request line is asserted by the 6010.

3.5 2420 Sample DMA Configuration Pseudo C Code

```
// Global Configuration Register
DMA4_GCR = 0x00xx0020;

// Channel Link Control register: disable since default is unknown.
DMA4_CLnkCtrl = 0x00000000;

// Disable interupts.
DMA4_CICR = 0x00000000;

// Channel Status Register: reset bits since default is unknown.
DMA4_CSR = 0x00000FFE;

// Channel Source Start Address: 6010 FIFO or 2420 system memory. Must be aligned on a 32-byte
burst boundary.
DMA4_CSSA = Source;

// Destination Start Address: 6010 FIFO or 2420 system memory. Must be aligned on a 32-byte burst
boundary.
DMA4_CDSA = Destination;

// Channel Control Register: use external DMA request.
qReg = Set the DMA4_CCRn SYNCHRO_CONTROL fields appropriately;

// Channel Element Number register: elements (quadlets) per frame.
// FrameSizeQuadlets = 4 for a 8 × 16 NOR burst, 8 for a 16 × 16 NOR burst
DMA4_CEN = FrameSizeQuadlets;

// Channel Frame Number register: Frames to be transferred. Need to round up.
// (DataSizeQuadlets = total bytes to transfer / 4)
CFN = ((DataSizeQuadlets + (FrameSizeQuadlets -1))/ FrameSizeQuadlets);

//NOR burst of 16 × 16
if ( 6010 FIFO -> 2420 Memory, 6010 FIFO READ )
{
// Source is Double Index address mode and DESTINATION is Post increment address mode.
qReg |= 0x01007020;
DMA4_CCR = qReg;

// Channel Source Destination Parameter register:
// Element type - 32 bits or quadlet.
// Source burst - 8 quadlets.
// Destination - 8 quadlets.
DMA4_CSDP = 0x00018102;

// Channel source element index.
DMA4_CSEI = 1;

// Channel source frame index. Address needs to be decremented by 32 bytes.
DMA4_CSFI = -31;

// Channel destination element index.
DMA4_CDEI = 0;

// Channel destination frame index.
DMA4_CDFI = 0;
}
else // 2420 Memory -> 6010 FIFO, 6010 FIFO WRITE
{
// Destination is Double Index address mode and SOURCE is Post increment address mode.
```

```
qReg |= 0x0080D020;
DMA4_CCR = qReg;

// Channel Source Destination Parameter register:
// Element type - 32 bits or quadlet.
// Source burst - 8 quadlets.
// Destination - 8 quadlets.
DMA4_CSDP = 0x00018102;

// Channel source element index.
DMA4_CSEI = 0;

// Channel source frame index.
DMA4_CSFI = 0;

// Channel destination element index.
DMA4_CDEI = 1;

// Channel destination frame index. Address needs to be decremented by 32 bytes.
DMA4_CDFI = -31;
}

// NOR burst of 8x16
if( 6010 FIFO -> 2420 Memory, 6010 FIFO READ )
{
// Source is Double Index address mode and DESTINATION is Post increment address mode.
qReg |= 0x01007020;
DMA4_CCR = qReg;

// Channel Source Destination Parameter register:
// Element type - 32 bits or quadlet.
// Source burst - 4 quadlets.
// Destination - 4 quadlets.
DMA4_CSDP = 0x00014082;

// Channel source element index.
DMA4_CSEI = 1;

// Channel source frame index. Address needs to be decremented by 16 bytes.
DMA4_CSFI = -15;

// Channel destination element index.
DMA4_CDEI = 0;

// Channel destination frame index.
DMA4_CDFI = 0;
}
else // 2420 Memory -> 6010 FIFO, 6010 FIFO WRITE
{// Destination is Double Index address mode and SOURCE is Post increment address mode.
qReg |= 0x0080D020;
DMA4_CCR = qReg;

// Channel Source Destination Parameter register:
// Element type - 32 bits or quadlet.
// Source burst - 4 quadlets.
// Destination - 4 quadlets.
DMA4_CSDP = 0x00014082;

// Channel source element index.
DMA4_CSEI = 0;

// Channel source frame index.
DMA4_CSFI = 0;

// Channel destination element index.
DMA4_CDEI = 1;

// Channel destination frame index. Address needs to be decremented by 16 bytes.
DMA4_CDFI = -15;
}
// Enable Channel
qReg = CCR;
qReg |= 0x00000080;
DMA4_CCR = qReg;
```

4 6010 DMA Configuration

4.1 Overview

When the 6010 is in DMA mode, it uses one of its six DMA request control lines to signal the 2420 that data is ready to be read on one of the end point FIFOs, or that the USB is ready for data to be written into one of the end point FIFOs.

4.2 DMA Request Setup

DMA request zero and one are on dedicated pins, they are also multiplexed on GPIO zero and one respectively. DMA request two through five are multiplexed on GPIO two through five. If a GPIO is to be used as a DMA request, the proper bit must be set in the GPIO Configuration Register[29:24]. Bit 29 corresponds to DMA request 5 and bit 24 corresponds to DMA request 0. The dedicated pins DMAREQ0/1 require no register configuration to work, they are always enabled. If an end point is to be serviced using the DMA, it must be mapped to one of the six DMA request lines. This is done in the DMA request endpoint mapping register. The DirDMAReqN bit fields are set to indicate whether the DMA request is a TX or RX end point. The DMAReqNEP bit fields indicate which end point FIFO is serviced using DMA request N (N refers to DMA request 5–0). For short packets (less than the values specified in USB IP IN/OUT EP #n max packet size register), the 6010 DMA request logic will not assert the external signal to the 2420. In this case, software must adjust the XfrSizeEpn field in the 6010 USB IP end point IN/OUT #n access configuration register and set the SwEnEpn bit. This will force the assertion of the DMA request signal to the 2420 and start the DMA.

4.3 NOR Interface Setup

The burstsize bit field of the DMA request configuration register must be set to match the burst size selected in the ATTACHED_DEVICE_PAGELENGTH bit field of the GPMC_CONFIG1_N register. For the best throughput, a burst size of 16×16 should be used. If a DMA request line is to be used, it must be enabled in the DmaReqEn bit field of the DMA requestconfiguration register[25:20]. Bit 25 corresponds to DMA request 5 and bit 20 corresponds to DMA request 0. The DmaReqAsser bit field can be left set to its default value of 0x0000. This will ensure the fastest possible turnaround time for the 2420 DMA. The WaitCount bit field in the device wait count register must be set to 0x1 in order to properly work with the 2420 DMA and NOR interface.

By default, there are internal pullup resistors on the DMA request lines. These pull-up resistors are enabled by default. Depending on the architecture of the target system, these can be disabled in the device pullup disable control registers if needed.

4.4 6010 Sample DMA Configuration Pseudo Code

```
// Configure the USB IP IN/OUT EP #n Max Packet Size Registers and TXMAXPn/RXMAXPn Registers
USB IP IN/OUT EP #n Max Packet Size Register = TXMAXPn/RXMAXPn Register = 512 (most likely this
will be 512
bytes for HS USB)

// Set Wait Count to 1 to work with GPMC Write
Device Wait Count Register = 0x00000001

// Enable the 6 DMA request lines if they are used on the GPIO.
GPIO Configuration Register = 0x3f000000

// Connect the DMA request lines to the appropriate end points
DMA request end point mapping register = system dependent

// Set end point transfer size
USB IP end point IN/OUT #n Access Configuration = total number of bytes to DMA
```

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information 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.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Clocks and Timers	www.ti.com/clocks	Digital Control	www.ti.com/digitalcontrol
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Telephony	www.ti.com/telephony
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265