



Please note that Cypress is an Infineon Technologies Company.

The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content

The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers

Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.



THIS SPEC IS OBSOLETE

Spec No: 001-15284

Spec Title: AN4051 - FX2LP(TM) GPIF FLOW STATE
FEATURE FOR UDMA

Replaced by: None

FX2LP™ GPIF Flow State Feature for UDMA

Author: Rich Peng
Associated Part Family: CY768xxx
Related Application Notes: None

To get the latest version of this application note, or the associated project file, please visit <http://www.cypress.com/go/AN4051>.

AN4051 introduces the FX2LPTM GPIF flow state feature, which extends GPIF to handle ATAPI UDMA. This application note includes flow state definitions and related register functions. It concludes with an example of how to use the flow state control in a UDMA transfer.

Contents

1	Introduction.....	1	3.15 HOCTL[2:0]	8
2	Flow State Definition.....	2	3.16 FALLING.....	8
3	Register Summary.....	3	3.17 RISING	8
	3.1 Flow State Registers.....	3	3.18 D[7:0].....	8
	3.2 FSE.....	3	3.19 General GPIF Registers	9
	3.3 FS[2:0]	3	3.20 HOLDTIME[1:0]	9
	3.4 LFUNC[1:0].....	4	3.21 UDMA CRC Registers	9
	3.5 TERMA/B[2:0].....	4	3.22 QENABLE.....	10
	3.6 CTLOE[3:0].....	5	3.23 QSTATE	10
	3.7 CTL[5:0].....	5	3.24 QSIGNAL.....	11
	3.8 SLAVE	6	3.25 Waveform Descriptor Enhancements	11
	3.9 RDYASYNC.....	6	3.26 SGL	11
	3.10 CTLTOGL	6	3.27 SGLCRC.....	11
	3.11 SUSTAIN	6	4 Summary.....	12
	3.12 MSTB[2:0].....	7	Appendix A. Example UDMA Waveform Descriptors	13
	3.13 HOPERIOD[3:0].....	7	A.1 UDMA IN Waveform Descriptors	13
	3.14 HOSTATE.....	7	A.2 UDMA OUT Waveform Descriptors	14

1 Introduction

Although the FX2LPTM GPIF flow state feature was created for UDMA, it is not limited to that interface. The GPIF flow state can capture other bus protocols, which gives it a value and lifetime that extends beyond UDMA. This is in keeping with the basic philosophy of GPIF.

In particular, the flow state was designed for UDMA mode 4 (66 MB/s) and mode 5 (100 MB/s). UDMA is an aggressive protocol in general, and these modes give the added difficulty of speed on top of that.

Some of the features of the UDMA protocol include:

- asynchronous
- double-edge data strobes
- host masters the data phase of OUTs (host-to-device)
- device masters the data phase of INs
- host or device data pausing
- host or device burst termination
- CRC calculation and reporting upon termination

Note: The term “host” refers to GPIF, and “device” means, for example, a disk drive

A very capable flow state feature was created thanks to these requirements and therefore may be able to serve other aggressive protocols encountered in a USB system. Although the flow state feature will be efficient in handling the more aggressive protocols, it can be used for more mundane protocols as well.

2 Flow State Definition

A typical data transaction involves a setup phase, a data phase, and a completion phase. The flow state feature is highly efficient at handling the data phase, especially if the data phase involves the transfer of multiple data samples. A flow state can handle the entire data phase while using up just one GPIF state. This leaves the other six states for handling the setup and completion phases of the transaction.

A flow state can be considered to be an extra layer on top of the normal decision point or interval states of the GPIF. The flow state does not replace these basic states; it only enhances them. Although the flow state can be an interval or a decision point, the most likely use would be as a decision point. This allows the GPIF to sense when the correct amount of data has been transferred so that it can branch to the completion phase of the transaction. If a flow state were set up as an interval, this would simply mean that the data phase would have a predictable and repeatable time duration and behavior. This is not likely in most multi-sample transactions.

The idea behind a flow state is to use one GPIF state to throttle data on the bus. The user sets up a flow logic register (FLOWLOGIC), similar to the RDY logic opcode in a waveform descriptor, to decide when to allow data to flow and when to cause it to temporarily halt. The flow logic can consider up to two flow logic sources including internal FIFO flags, the transaction count expiration flag, any of the six RDY pins, and an 8051 RDY signal.

The flow state feature allows the GPIF to be set up as a master or as a slave on the bus. The GPIF can be the one creating the read and write signals, or an external master can perform those functions. For example, for UDMA OUT transactions, the GPIF is the master of the data transfer. For UDMA IN, the device is the master and the GPIF is the slave.

In the case where the GPIF is the master, the data flow will work in the following way:

- The user will pick one of the CTL[5:0] pins to be the master strobe signal that controls data transfer.
- The user will program how fast the master strobe can toggle and whether data is transferred between the internal FIFOs and the data bus on the rising, falling, or double-edge of the master strobe.
- The user will then program the flow logic to control when the master strobe toggles or freezes, thus controlling the flow of data without having to leave the GPIF flow state.

In the case where the GPIF is a slave to an external master, the data flow will work nearly the opposite way:

- The user tells the GPIF which of the six RDY pins is the master strobe from the external master.
- The user will then program the flow logic to control the behavior of the six CTL pins. This allows the user to assert a ready signal to the external master when the flow logic evaluates one way, and assert not ready when the flow logic evaluates the opposite way. Once again, this controls the flow of data without having to leave the GPIF flow state.

In either case, whether the GPIF is the master or the slave during the flow state, the data flow mechanism will carry on indefinitely until the decision point logic detects that it is time to branch out to the completion phase of the transaction. The decision point logic can use an entirely different pair of ready sources and logic function than what was used for the flow logic.

There are eight registers that define a flow state. Not all of them need to be set up for any particular flow state. Five of the eight registers must be set up:

- FLOWSTATE
- FLOWLOGIC
- FLOWEQ0CTL
- FLOWEQ1CTL
- FLOWSTB

The other three registers are programmed only if the situation calls for it and typically only once if so: FLOWHOLDOFF, FLOWSTBEDGE, and FLOWSTBHPERIOD. See the “GPIF Flow State Registers” section of the [FX2LP Technical Reference Manual](#) for further explanation of these registers.

In addition to these core eight registers, a general GPIF register called GPIFHOLDAMOUNT has been added as well as three UDMA CRC related registers: UDMACRCH, UDMACRCL, and UDMACRCQUALIFIER. The CRC registers typically do not need attention from the 8051 since this calculation is handled automatically by the GPIF. See the [FX2LP Technical Reference Manual](#) for further explanation of these registers.

The final enhancements associated with the new flow state are in the waveform descriptor itself. A bit was added to the ACTION opcode that allows the GPIF to do a single (as in SGLDATH/L) transaction within a FIFO waveform. An additional enhancement is in the CTL opcode of the waveform descriptor, which carries extra meaning if the SUSTAIN bit (FLOWSTB.4) is set. See the FLOWSTB register discussion for further explanation of these two new enhancements.

For example, waveform descriptors that can handle Mode 4 or Mode 5 UDMA IN or OUT transactions, see Appendix A.

3 Register Summary

The Register Summary is broken into four subsections: Flow State Registers, General GPIF Registers, UDMA CRC Registers, and Waveform Descriptor Enhancements. This section summarizes all of the registers and waveform descriptors that are associated with the flow state feature.

Each register has its own table consisting of eight rows showing the register name and address in the first row, the name of the bits in the second row, the default values in the third row, and the suggested settings for UDMA mode 4 IN, mode 4 OUT, mode 5 IN, and mode 5 OUT in rows 4-7, respectively. The eighth and final row shows the 8051's Read/Write accessibility to the individual bits in the register.

3.1 Flow State Registers

FLOWSTATE						E6C6		
Bits 7:0	FSE	0	0	0	0	FS[2:0]		
Default	0	0	0	0	0	0	0	0
M4IN	1	0	0	0	0	depends on waveform		
M4OUT	1	0	0	0	0	depends on waveform		
M5IN	1	0	0	0	0	depends on waveform		
M5OUT	1	0	0	0	0	depends on waveform		
Access	RW	R	R	R	R	RW	RW	RW

Any one (and only one) of the seven GPIF states in a waveform can be programmed to be the flow state. This register defines which state, if any, in the next invoked GPIF waveform will be the flow state.

3.2 FSE

Global enable for the flow state. When it is disabled, all flow state registers are set to “don’t care”, and the next waveform invocation will not cause a flow state to be used.

3.3 FS[2:0]

Defines which GPIF state is the flow state. Valid values are 0-6.

FLOWLOGIC					E6C7			
Bits 7:0	LFUNC[1:0]		TERMA[2:0]			TERMB[2:0]		
Default	0	0	0	0	0	0	0	0
M4IN	0	0	FIFO PF as an Almost Full			FIFO PF as an Almost Full		
M4OUT	0	1	FIFO PF as an Almost Empty			RDY pin that is DDMARDY#		
M5IN	0	0	FIFO PF as an Almost Full			FIFO PF as an Almost Full		
M5OUT	0	1	FIFO PF as an Almost Empty			RDY pin that is DDMARDY#		
Access	RW	RW	RW	RW	RW	RW	RW	RW

The bit definitions for this register are analogous to the bit definitions in the RDY LOGIC opcode in a waveform descriptor. However, instead of controlling the branching for a decision point, it controls the freezing or flowing of data on the bus in a flow state.

The user defines the states of CTL[5:0] for when the flow logic equals 0 and 1 (see FLOWEQ0CTL and FLOWEQ1CTL). This is useful in activating or deactivating protocol ready signals to hold off an external master (where the GPIF is acting like a slave) in response to internal FIFO flags warning of an impending underflow or overflow situation.

In the case where the GPIF is the master, the user also defines whether MSTB (a CTL pin in this case; see FLOWSTB) toggles (reads or writes data on the bus) when the flow logic evaluates to a 1 or a 0. This is useful for the GPIF to consider protocol ready signals from the slave as well as FIFO flags to decide when to clock data out of or into the FIFOs and when to freeze the data flow instead.

It should be noted that this flow logic does not replace the decision point logic defined in a waveform descriptor. The decision point logic in a waveform descriptor is still used to decide when to branch out of the flow state. The decision point logic can use an entirely different pair of ready signals than the flow logic in making its decisions.

3.4 LFUNC[1:0]

00 = A AND B
 01 = A OR B
 10 = A XOR B
 11 = !A AND B

Because the flow logic decision can be based on the output being a 1 or a 0, NAND, NOR, XNOR, and !(A AND B), operations can be achieved as well. Note that !(A AND B) is the same as (A OR !B).

Note: The logic function !(A AND B) has been added to the list of choices for the RDY LOGIC opcode in a waveform descriptor as well; previously, setting 11 was reserved.

3.5 TERMA/B[2:0]

0 = RDY[0]
 1 = RDY[1]
 2 = RDY[2]
 3 = RDY[3]
 4 = RDY[4]
 5 = RDY[5] or TCXRDY5 (depending on GPIFREADYCFG.5)
 6 = FIFO Flag (PF, EF, or FF depending on GPIFEPxFLAGSEL)
 7 = 8051 RDY (GPIFREADYCFG.7)

FLOWEQ0CTL					E6C8			
Bits 7:0	CTLOE3	CTLOE2	CTLOE1/CTL5	CTLOE0/CTL4	CTL3	CTL2	CTL1	CTL0
Default	0	0	0	0	0	0	0	0
M4IN	CTL pin that is HDMARDY# is 0 (based on suggested FLOWLOGIC settings) CTL pin that is STOP is 0; CTL pin that is DMACK# is 0							
M4OUT	CTL pin that is HSTROBE is don't care (taken care of in the 3 FLOWSTB registers) CTL pin that is STOP is 0; CTL pin that is DMACK# is 0							
M5IN	CTL pin that is HDMARDY# is 0 (based on suggested FLOWLOGIC settings) CTL pin that is STOP is 0; CTL pin that is DMACK# is 0							
M5OUT	CTL pin that is HSTROBE is don't care (taken care of in the 3 FLOWSTB registers) CTL pin that is STOP is 0; CTL pin that is DMACK# is 0							
Access	RW	RW	RW	RW	RW	RW	RW	RW

FLOWEQ1CTL					E6C9			
Bits 7:0	CTLOE3	CTLOE2	CTLOE1/CTL5	CTLOE0/CTL4	CTL3	CTL2	CTL1	CTL0
Default	0	0	0	0	0	0	0	0
M4IN	CTL pin that is HDMARDY# is 1 (based on suggested FLOWLOGIC settings) CTL pin that is STOP is 0; CTL pin that is DMACK# is 0							
M4OUT	CTL pin that is HSTROBE is don't care (taken care of in the 3 FLOWSTB registers) CTL pin that is STOP is 0; CTL pin that is DMACK# is 0							
M5IN	CTL pin that is HDMARDY# is 1 (based on suggested FLOWLOGIC settings) CTL pin that is STOP is 0; CTL pin that is DMACK# is 0							
M5OUT	CTL pin that is HSTROBE is don't care (taken care of in the 3 FLOWSTB registers) CTL pin that is STOP is 0; CTL pin that is DMACK# is 0							
Access	RW	RW	RW	RW	RW	RW	RW	RW

These two registers define the state of CTL[5:0] when the output of the flow logic (see FLOWLOGIC) equals 0 and when it equals 1. Their definition is analogous to the CTL opcode in a waveform descriptor. During a flow state, the CTL opcode in the waveform descriptor is completely ignored and the behavior of the CTL[5:0] pins are defined by these two registers instead.

3.6 CTLOE[3:0]

If GPIFCTLOUTCFG.7 = 1, then these bits become the tristate enables for CTL[3:0], and CTL[5:4] are ignored.

3.7 CTL[5:0]

If GPIFCTLOUTCFG.7 = 1 then:

- Bits 7:4 become CTLOE[3:0] and the CTL[5:4] pins are unused.
 - Bits 3:0 define the CMOS levels of the CTL[3:0] pins if the corresponding CTLOE bit is 1, else they will be tristate.
- Else,
- Bits 5:0 become the CMOS levels of the CTL[5:0] pins unless the corresponding bits in GPIFCTLOUTCFG[5:0] cause these pins to be open-drain; in which case, a 1 means tristate and a 0 means drive 0.

FLOWSTB					E6CB	
Bits 7:0	SLAVE	RDYASYNC	CTLTOGL	SUSTAIN	0	MSTB[2:0]

Default	0	0	1	0	0	0	0	0
M4IN	1	1	don't care	don't care	0	RDY pin that is DSTROBE		
M4OUT	0	don't care	0*	1	0	CTL pin that is HSTROBE		
M5IN	1	1	don't care	don't care	0	RDY pin that is DSTROBE		
M5OUT	0	don't care	0*	1	0	CTL pin that is HSTROBE		
Access	RW	RW	RW	RW	R	RW	RW	RW

* - based on suggested FLOWLOGIC settings

This register defines the master signal, called MSTB that causes data to be read or written during a flow state.

For transactions where GPIF is the slave on the bus, MSTB will be one of the RDY[5:0] pins. This includes external masters that can either write data into GPIF (for example, UDMA IN) or read data out of GPIF.

For transactions where GPIF is the master on the bus, MSTB will be one of the CTL[5:0] pins. This includes cases where the GPIF writes data out to a slave (for example, UDMA OUT) or reads data from a slave.

If MSTB is a CTL pin, the user must also define the rate at which the CTL pin toggles (see FLOWSTBHPERIOD). The CTL pin will toggle only when the output of the flow logic equals the value defined in the CTLTOGL bit. Otherwise, the CTL pin will freeze. During a flow state, the behavior of the CTL pins is normally defined by the FLOWEQ0/1CTL registers, not the CTL opcode of the waveform descriptor. The exception to this is the MSTB CTL pin (which only exists if the GPIF is the master on the bus for the transaction) whose behavior is described by the following four registers: FLOWLOGIC, FLOWSTB, FLOWSTBEDGE, and FLOWSTBHPERIOD.

It should be noted that in the case where MSTB is a CTL pin, the GPIF will produce at most one extra active edge on MSTB in response to a not ready indication from the slave. This is regardless of whether the RDY pin is asynchronous. Also, this is critical to UDMA OUT, which requires that no more than three extra words be written by the host (the GPIF) in response to the deactivation of DDMARDY#.

An additional consideration if MSTB is a CTL pin, is whether it is used to write data into a slave. If this is the case, the user must define how long to hold the data with respect to the activating CTL pin edge (see GPIFHOLDAMOUNT).

Regardless of whether MSTB is a CTL or a RDY pin, the user must define which edge — rising, falling, or double-edge — will cause a data read or write (see FLOWSTBEDGE).

3.8 SLAVE

0: GPIF is the master of the bus transaction. This means that one of the CTL[5:0] pins will be the MSTB and the particular one is selected by MSTB[2:0].

1: GPIF is the slave of the bus transaction. This means that one of the RDY[5:0] pins will be the MSTB and the particular one is selected by MSTB[2:0].

3.9 RDYASYNC

If SLAVE is 0 then this bit is ignored, otherwise:

0: MSTB (which is a RDY pin in this case) is asynchronous to IFCLK

1: MSTB (which is a RDY pin in this case) is synchronous to IFCLK

3.10 CTLTOGL

If SLAVE is 1, this bit is ignored. Otherwise, this bit defines which state of the flow logic (see FLOWLOGIC) causes MSTB (which will be a CTL pin in this case) to toggle. For example, if this bit is set to 1, then if the output of the flow logic equals 1, then MSTB will toggle, which causes data to flow on the bus. If in the same example the output of the flow logic equals 0, then MSTB will freeze, which causes data flow to halt on the bus.

3.11 SUSTAIN

If SLAVE is 1, this bit is ignored. Although this bit has a very specific value to UDMA, it is described here in general terms in the unlikely scenario that another protocol could use its capability.

Upon exiting a flow state in which SLAVE is 0, MSTB (which is a CTL pin in this case) will normally go back to adhering to the CTL opcodes defined in the waveform descriptor. However, for some protocols (UDMA), it is necessary to freeze MSTB at the value it was when the flow state exited. Furthermore, it may be desirable at some point later in the protocol (UDMA) to return MSTB to a 1 regardless of what it had been frozen to.

It may be impossible for the waveform designer to predict what state MSTB will be in when the flow state exits (due to not knowing how many pieces of data will be transferred a priori in a double-edge system for example; UDMA), and thus may not be able to accurately describe the desired states of MSTB upon completing the waveform.

This bit will relieve the user from having to make this prediction. With this bit set, the bit in the CTL opcode that corresponds to the MSTB signal will take on the following new meaning. For states immediately following a flow state, if the waveform bit is 0, then it means sustain the value of MSTB. If the waveform bit is 1 then it means return MSTB to a 1 regardless of its prior state. After the waveform bit is set to 1, then the sustain feature is released and the CTL opcode bit returns to its normal definition for the remaining states in the waveform.

3.12 MSTB[2:0]

If SLAVE is 0, these bits will select which CTL[5:0] pin is the MSTB. If SLAVE is 1, these bits will select which RDY[5:0] pin is the MSTB.

FLOWHOLDOFF					E6CA			
Bits 7:0	HOPERIOD[3:0]				HOSTATE	HOCTL[2:0]		
Default	0	0	0	0	0	0	0	0
M4IN	don't care	don't care	don't care	don't care	don't care	don't care	don't care	don't care
M4OUT	don't care	don't care	don't care	don't care	don't care	don't care	don't care	don't care
M5IN	0	0	1	0	1	CTL pin that is HDMARDY#		
M5OUT	don't care	don't care	don't care	don't care	don't care	don't care	don't care	don't care
Access	RW	RW	RW	RW	RW	RW	RW	RW

For flow state transactions that meet the following criteria:

1. The interface is asynchronous
2. GPIF is acting like a slave (FLOWSTB.7 = 1), and thus MSTB is a RDY pin
3. Data is being written into the GPIF
4. The rate at which data is being written in exceeds 96 MB/s for a word-wide data bus or 48 MB/s for a byte-wide data bus

This register is programmed to protect the synchronization interface from overflowing (micro-level). This is not to be confused with protecting the endpoint FIFOs from overflowing (macro-level), which is managed by the FLOWLOGIC register. If a transaction does not meet the above criteria, this register can be ignored because FX2LP is capable of synchronizing data rates up to 96 MB/s without getting backed up.

An example transaction that meets the above criteria would be mode 5 UDMA IN which can be as fast as 105 MB/s. Note, however, that if the UDMA device (for example, the disk drive) supports mode 5 UDMA but does not source data faster than 96 MB/s, then this register can be ignored.

3.13 HOPERIOD[3:0]

Defines how many IFCLK cycles to assert not ready (HOCTL) to the external master in order to allow the synchronization interface to catch up.

3.14 HOSTATE

Defines what the state of the HOCTL signal should be in to assert not ready.

3.15 HOCTL[2:0]

Defines which of the six CTL[5:0] pins will be the HOCTL signal which asserts not ready to the external master when the synchronization detects a potential overflow coming. It should coincide with the CTL[5:0] pin that is picked as the “not ready” signal for the (macro-level) endpoint FIFO overflow protection.

FLOWSTBEDGE					E6CC			
Bits 7:0	0	0	0	0	0	0	FALLING	RISING
Default	0	0	0	0	0	0	0	1
M4IN	0	0	0	0	0	0	1	1
M4OUT	0	0	0	0	0	0	1	1
M5IN	0	0	0	0	0	0	1	1
M5OUT	0	0	0	0	0	0	1	1
Access	R	R	R	R	R	R	RW	RW

This register defines whether MSTB (see FLOWSTB) causes data to read or be written on either the falling edge, the rising edge, or both (double-edge).

3.16 FALLING

0: data is not transferred on the falling edge of MSTB

1: data is transferred on the falling edge of MSTB

3.17 RISING

0: data is not transferred on the rising edge of MSTB

1: data is transferred on the rising edge of MSTB

Note To cause data to transfer on both edges of MSTB, set both bits to 1.

FLOWSTBPERIOD					E6CD			
Bits 7:0					D[7:0]			
Default	0	0	0	0	0	0	1	0
M4IN	0	0	0	0	0	0	don't care	don't care
M4OUT	0	0	0	0	0	0	1	1
M5IN	0	0	0	0	0	0	don't care	don't care
M5OUT	0	0	0	0	0	0	1	0
Access	RW	RW	RW	RW	RW	RW	RW	RW

If the flow state is such that the GPIF is the master on the bus (FLOWSTB.7 = 0), then MSTB will be one of the CTL[5:0] pins (see FLOWSTB). While in the flow state, if the flow logic (see FLOWLOGIC) evaluates in such a way that MSTB should toggle (see FLOWSTB.5), then this register defines the frequency at which it will toggle.

More precisely, the user defines the half period of the MSTB toggling frequency. Further, to give the user a high degree of resolution this MSTB half period is defined in terms of half IFCLK periods. Therefore, if IFCLK is running at 48 MHz this gives the user a resolution of 10.415 ns.

For example, mode 4 UDMA OUT transactions suggest that the half period of MSTB is 30 ns (to get 66 MB/s). Therefore, the user would write a 3 to this register (assuming 48 MHz IFCLK), thus obtaining a half period of 31.245 ns. This will give a transfer rate of 62.49 MB/s.

3.18 D[7:0]

Number of half IFCLK periods that define the half period of MSTB (if it is a CTL pin). The value must be at least 2, meaning that the minimum half period for MSTB is one full IFCLK cycle.

3.19 General GPIF Registers

GPIFHOLDAMOUNT					E60C			
Bits 7:0	0	0	0	0	0	0	HOLDTIME[1:0]	
Default	0	0	0	0	0	0	0	0
M4IN	0	0	0	0	0	0	don't care	don't care
M4OUT	0	0	0	0	0	0	0	1
M5IN	0	0	0	0	0	0	don't care	don't care
M5OUT	0	0	0	0	0	0	0	1
Access	RW	RW	RW	RW	RW	RW	RW	RW

For any transaction where the GPIF writes data onto FD[15:0], this register determines how long the data is held. Valid choices are 0, ½, or 1 IFCLK cycle. This register applies to any data written by the GPIF to FD[15:0], whether through a flow state or not.

For non-flow states, the hold amount is really just a delay of the normal (non-held) presentation of FD[15:0] by the amount specified in HOLDTIME[1:0].

For flow states in which the GPIF is the master on the bus (FLOWSTB.7 = 0), the hold amount is with respect to the activating edge (see FLOWSTBEDGE) of MSTB (which will be a CTL pin in this case).

For flow states in which the GPIF is the slave on the bus (FLOWSTB.7 = 1), the hold amount is really just a delay of the normal (non-held) presentation of FD[15:0] by the amount specified in HOLDTIME[1:0] in reaction to the activating edge of MSTB (which will be a RDY pin in this case). Note the hold amount is not directly with respect to the activating edge of MSTB in this case. It is with respect to when the data would normally come out in response to MSTB including any latency to synchronize MSTB.

In all cases, the data will be held for the desired amount even if the ensuing GPIF state calls for the data bus to be tristated. In other words, the FD[15:0] output enable will be held by the same amount as the data itself.

3.20 HOLDTIME[1:0]

00 = 0 IFCLK cycles
 01 = ½ IFCLK cycle
 10 = 1 IFCLK cycle
 11 = reserved

3.21 UDMA CRC Registers

UDMACRCH					E67D			
Bits 7:0	CRC[15:8]							
Default	0	1	0	0	1	0	1	0
M4IN	don't write	don't write	don't write	don't write	don't write	don't write	don't write	don't write
M4OUT	don't write	don't write	don't write	don't write	don't write	don't write	don't write	don't write
M5IN	don't write	don't write	don't write	don't write	don't write	don't write	don't write	don't write
M5OUT	don't write	don't write	don't write	don't write	don't write	don't write	don't write	don't write
Access	RW	RW	RW	RW	RW	RW	RW	RW

UDMACRCL					E67E			
Bits 7:0	CRC[7:0]							
Default	1	0	1	1	1	0	1	0
M4IN	don't write	don't write	don't write	don't write	don't write	don't write	don't write	don't write

M4OUT	don't write	don't write	don't write	don't write	don't write	don't write	don't write	don't write
M5IN	don't write	don't write	don't write	don't write	don't write	don't write	don't write	don't write
M5OUT	don't write	don't write	don't write	don't write	don't write	don't write	don't write	don't write
Access	RW	RW	RW	RW	RW	RW	RW	RW

These two registers are strictly for debug purposes. The CRC represented by these registers is calculated based on the rules defined in the ATAPI specification for UDMA transfers. It is calculated automatically by the GPIF as data is transferred on FD[15:0].

These registers will return the live calculation of the CRC at any point in the transfer, but will be reset to the seed value of 0x4ABA upon the GPIF entering the IDLE state. These registers are writeable by the 8051 and thus the currently calculated CRC including the seed value can be overwritten at any time. For modes 4 and 5, the 8051 should not write to these registers.

UDMACRCQUALIFIER					E67F			
Bits 7:0	QENABLE	0	0	0	QSTATE	Q SIGNAL		
Default	0	0	0	0	0	0	0	0
M4IN	1 if host terminated	0	0	0	0	CTL pin that is STOP		
M4OUT	0	0	0	0	don't care	don't care	don't care	don't care
M5IN	1 if host terminated	0	0	0	0	CTL pin that is STOP		
M5OUT	0	0	0	0	don't care	don't care	don't care	don't care
Access	RW	RW	RW	RW	RW	RW	RW	RW

This register only applies to UDMA IN transactions that are host terminated. Otherwise, this register can be completely ignored.

This register covers a very specific and potentially nonexistent (from a typical system implementation standpoint*) UDMA CRC situation. However rare the situation may be, it is still allowed by the ATAPI specification and thus must be considered and solved by this register.

The ATAPI specification says that if the host (in this case the GPIF) terminates a UDMA IN transaction, that the device (for example, the disk drive) is allowed to send up to three more words after the host deactivates the HDMARDY signal. These "dribble" words may not be of interest to the host and thus may be discarded. However, CRC must still be calculated on them, because the host and the device must compare and match the CRC at the end of the transaction to consider the transfer error-free.

The GPIF normally only calculates CRC on words that are written into the FIFO (and not discarded). This poses a problem because in this case some words will be discarded but still must be included in the CRC calculation. This register gives a way to have the GPIF calculate CRC on the extra discarded words as well.

The user would program this register in the following way. The QENABLE bit would be set to 1. The Q SIGNAL field would be programmed to select the CTL pin that coincides with the UDMA STOP signal. The QSTATE bit would be programmed to be 0. This would tell the GPIF to calculate CRC on any DSTROBE edges from the device when STOP=0, which solves the problem.

* - A typical UDMA system will have the device, instead of the host, terminate UDMA IN transfers thus completely avoiding this situation.

3.22 QENABLE

This bit enables the CRC qualifier feature, and thus the other bits in this register.

3.23 QSTATE

This bit says what state the CRC qualifier signal (selected by Q SIGNAL below) must be in to allow CRC to be calculated on words being written into the GPIF.

3.24 Q SIGNAL

These bits select which of the CTL[5:0] pins is the CRC qualifier signal.

3.25 Waveform Descriptor Enhancements

ACTION OPCODE					Waveform Descriptor Offset Bytes 8-14			
Bits 7:0	0	0	SGL	GINT	INCAD	NEXT/SGLCRC	DATA	DP
Default	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
M4IN	0	0	1*	0	0	1	1	0
M4OUT	0	0	1*	0	0	1	1	0
M5IN	0	0	1*	0	0	1	1	0
M5OUT	0	0	1*	0	0	1	1	0
Access	RW	RW	RW	RW	RW	RW	RW	RW

* - These settings only apply to the states in which the CRC will be driven on the bus.

The SGL bit allows the GPIF to present data from or sample data into the GPIFSGLDATH/LX registers during a FIFO type waveform. Or, alternatively the contents of the UDMACRCH/L registers can be presented on the bus. Any state can be programmed to do a single data transaction.

If the SGL bit is 1 then NEXT takes on new meaning (because NEXT has no use in a state where a single data transaction is occurring). The NEXT bit becomes redefined as the SGLCRC bit. If the SGLCRC bit is 1 then the contents of the UDMACRCH/L registers are presented on the bus (provided the DATA bit is 1 as well, which in this case means drive the data bus).

If, on the other hand, the SGLCRC bit is 0 then the GPIFSGLDATAH/L registers will be used in the transaction instead of the UDMACRCH/L registers. If DATA is 1, the contents of the GPIFSGLDATH/LX registers will be driven on the bus. If DATA is 0, then the bus is sampled and the results are placed in the GPIFSGLDATH/LX registers for the 8051 to read.

This feature only applies to FIFO type waveform invocations (such as GPIFTRIG invocations). For single data type waveform invocations (such as GPIFSGLDATH/LX invocations), this feature does not apply because the transfer of the GPIFSGLDATH/LX registers is taken care of by the fact that it was a single data invocation in the first place.

3.26 SGL

0: This particular state is not a single data transaction within a FIFO waveform

1: This particular state is a single data transaction within a FIFO waveform; NEXT becomes redefined as SGLCRC

3.27 SGLCRC

This bit only has meaning if SGL = 1.

0: The data that gets driven onto the bus (DATA = 1) or sampled from the bus (DATA = 0) will come from/go to the GPIFSGLDATH/LX registers

1: The data that gets driven onto the bus (DATA must be 1) will come from the UDMACRCH/L registers.

CTL OPCODE					Waveform Descriptor Offset Bytes 16-22			
Bits 7:0	CTLOE3	CTLOE2	CTLOE1/CTL5	CTLOE0/CTL4	CTL3	CTL2	CTL1	CTL0
Default	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
M4IN	normal definition							
M4OUT	normal definition except for the CTL pin that is HSTROBE							
M5IN	normal definition							
M5OUT	normal definition except for the CTL pin that is HSTROBE							
Access	RW	RW	RW	RW	RW	RW	RW	RW

If a waveform involves a flow state (FLOWSTATE.7 = 1), and the GPIF is the master on the bus (FLOWSTB.7 = 0), and the SUSTAIN bit is set (FLOWSTB.4 = 1) then the CTL bit of this OPCODE that corresponds to the MSTB CTL pin (FLOWSTB2:0) is redefined from its normal function.

For states immediately following a flow state, if the specified CTL bit is 0, it means sustain the value of MSTB. If the specified CTL bit is 1, it means return MSTB to a 1 regardless of its prior state. After the specified CTL bit is 1, then the sustain feature is released and the CTL opcode bit returns to its normal definition for the remaining states in the waveform.

For more information on this feature, see the definition of the SUSTAIN bit for the FLOWSTB register.

4 Summary

This document gives the details on how to enable GPIF Flow State Feature with ATAPI UDMA control. If your application needs to use ATAPI UDMA mode to enhance the speed then this example exactly address what you need. If the application in your design is not for the APATI UDMA control then this document still able to give you an example how to enable the GPIF Flow State Feature. In this document, it gives the details in each register usage and how to handle the flow state in each data transition. It also gives the detail about UDMA waveform setup in [Appendix A](#) which is a real example to make you able to completely understand the usage of GPIF Flow State Feature.

About the Author

Name: Rich Peng
Title: Product Apps Group Lead

Appendix A. Example UDMA Waveform Descriptors

The following section shows example waveform descriptors for various UDMA scenarios. These waveform descriptors have been tested and proven in simulation. There are six different waveform descriptors broken into two main categories: UDMA IN and UDMA out. Within each category there are three different waveforms representing whether the descriptor handles: host termination, device termination, or both host and device termination. The most generic descriptors are those that handle both host and device termination, but all three are shown for completeness.

The waveforms were designed to handle Mode 4. The exact same waveforms handle Mode 5 as well. The difference comes in how the flow state registers described in the Register Summary are set up. See that section for the suggested settings for Mode 4 and Mode 5, IN or OUT.

The bold state in each table denotes the flow state for that descriptor.

A.1 UDMA IN Waveform Descriptors

UDMA IN Supporting Host Termination							
LENGTH / BRANCH		OPCODE		OUTPUT		LOGIC FUNCTION	
Offset 0-6	val	Offset 8-14	val	Offset 16-22	val	Offset 24-30	val
State 0	08	State 0	01	State 0	07	State 0	00
State 1	01	State 1	00	State 1	06	State 1	00
State 2	13	State 2	01	State 2	00	State 2	E8
State 3	05	State 3	00	State 3	02	State 3	00
State 4	25	State 4	01	State 4	06	State 4	00
State 5	01	State 5	26	State 5	06	State 5	00
State 6	01	State 6	26	State 6	07	State 6	00

UDMA IN Supporting Device Termination							
LENGTH / BRANCH		OPCODE		OUTPUT		LOGIC FUNCTION	
Offset 0-6	val	Offset 8-14	val	Offset 16-22	val	Offset 24-30	val
State 0	08	State 0	01	State 0	07	State 0	00
State 1	01	State 1	00	State 1	06	State 1	00
State 2	13	State 2	01	State 2	00	State 2	00
State 3	01	State 3	26	State 3	06	State 3	00
State 4	3F	State 4	27	State 4	07	State 4	3F
State 5	00	State 5	00	State 5	FF	State 5	00
State 6	00	State 6	00	State 6	FF	State 6	00

UDMA IN Supporting both Host and Device Termination*							
LENGTH / BRANCH		OPCODE		OUTPUT		LOGIC FUNCTION	
Offset 0-6	val	Offset 8-14	val	Offset 16-22	val	Offset 24-30	val
State 0	08	State 0	01	State 0	07	State 0	00
State 1	01	State 1	00	State 1	06	State 1	00
State 2	13	State 2	01	State 2	00	State 2	E8
State 3	05	State 3	00	State 3	02	State 3	00
State 4	25	State 4	01	State 4	06	State 4	00
State 5	01	State 5	26	State 5	06	State 5	00
State 6	01	State 6	26	State 6	07	State 6	00

* - This waveform descriptor is exactly the same as the "UDMA IN Supporting Host Termination" descriptor above. It is duplicated here for completeness.

A.2 UDMA OUT Waveform Descriptors

UDMA OUT Supporting Host Termination							
LENGTH / BRANCH		OPCODE		OUTPUT		LOGIC FUNCTION	
Offset 0-6	val	Offset 8-14	val	Offset 16-22	val	Offset 24-30	val
State 0	08	State 0	01	State 0	07	State 0	00
State 1	01	State 1	00	State 1	06	State 1	00
State 2	05	State 2	02	State 2	02	State 2	00
State 3	23	State 3	03	State 3	02	State 3	2D
State 4	03	State 4	02	State 4	00	State 4	00
State 5	2E	State 5	03	State 5	04	State 5	00
State 6	01	State 6	26	State 6	06	State 6	00

UDMA OUT Supporting Device Termination							
LENGTH / BRANCH		OPCODE		OUTPUT		LOGIC FUNCTION	
Offset 0-6	val	Offset 8-14	val	Offset 16-22	val	Offset 24-30	val
State 0	08	State 0	01	State 0	07	State 0	00
State 1	01	State 1	00	State 1	06	State 1	00
State 2	05	State 2	02	State 2	02	State 2	00
State 3	1C	State 3	03	State 3	02	State 3	00
State 4	02	State 4	26	State 4	06	State 4	00
State 5	3F	State 5	27	State 5	07	State 5	3F
State 6	00	State 6	00	State 6	00	State 6	00

UDMA OUT Supporting both Host and Device Termination*							
LENGTH / BRANCH		OPCODE		OUTPUT		LOGIC FUNCTION	
Offset 0-6	val	Offset 8-14	val	Offset 16-22	val	Offset 24-30	val
State 0	08	State 0	01	State 0	07	State 0	00
State 1	01	State 1	00	State 1	06	State 1	00
State 2	05	State 2	02	State 2	02	State 2	00
State 3	1C	State 3	03	State 3	02	State 3	E8
State 4	03	State 4	02	State 4	00	State 4	00
State 5	2E	State 5	03	State 5	04	State 5	00
State 6	01	State 6	26	State 6	06	State 6	00

* - This waveform descriptor is exactly the same as the "UDMA OUT Supporting Host Termination" descriptor above with the exception of the two italicized bytes.

Document History

Document Title: AN4051 - FX2LP™ GPIF Flow State Feature for UDMA

Document Number: 001-15284

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	1736163	KUH	11/13/2007	Obtained spec # to add to the spec system.
*A	3177111	LIP	02/18/2011	Updated links and replaced FX2 with FX2LP.
*B	3252242	LIP	05/08/2011	Updated Abstract and removed obsolete links.
*C	3719073	LIP	08/21/2012	Added Summary. Updated in new template.
*D	4826096	LIP	07/07/2015	Updated the document as per new template.
*E	5807001	AESATP12	07/10/2017	Updated logo and copyright.
*F	5890087	RYLE	09/19/2017	Sunset review. Removed repeated text in Introduction. Obsoleted and published to web.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at [Cypress Locations](#).

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#) | [PSoC 6](#)

Cypress Developer Community

[Forums](#) | [WICED IOT Forums](#) | [Projects](#) | [Videos](#) | [Blogs](#) | [Training](#) | [Components](#)

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.



Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2007-2017. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spanion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spanion, the Spanion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.