

Scope

The intent of this document is to describe the additional features of GPIF that were added to support high speed interfaces like UDMA. In general “Host” will refer to the FX2 and Device will refer to the mass storage device or Hard disk. It is assumed that the reader is familiar with the ATAPI-6 specification (<http://www.t13.org/project/d1410r1b.pdf>) and familiar with the FX2 architecture and operation in general. The GPIF waveform tool V2.0 or later must be used in order to support the new GPIF features.

Overview

UDMA is an extension of the existing IDE interface that enables data transfer rates up to 100MBPS. UDMA is an asynchronous interface with the data strobe being driven by the device generating the data. In the case of data reads from the drive, the drive generates the data strobe along with the data being clocked on both edges of the data strobe. UDMA also requires a CRC for each transfer burst. The host calculates a CRC value for each word of data transferred into or out of the host controller. At the end of the UDMA transfer burst the CRC value is driven onto the data bus, by the host controller, for the drive to read.

Existing signals on the ATAPI interface have been redefined to support UDMA as follows:

<u>Current definition</u>	<u>UDMA definition</u>
DIOR	HSTROBE/HDMARDY
DIOW	STOP
IORDY	DSTROBE/DDMARDY

Note: Reads and writes are Host centric

Signal	Source	Function
DMAREQ	Device	Signals start/end of UDMA transfer
HSTROBE/HDMARDY	Host	Writes, used as data strobe/Reads, Throttles data from device
DSTROBE/DDMARDY	Device	Writes, used by device to throttle host/Reads, data strobe
DMACK	Host	Used to acknowledge UDMA request and to strobe CRC and end of cycle

Table 1 UDMA signal definition

UDMA transfers can be broken into three basic phases, burst initiation, data burst phase, and burst termination. The initiation and termination phases are managed by individual GPIF states. The data burst phase is considered to be one GPIF state but is also defined as a flow state. The flow state manages data flow and data throttle conditions. The following is an example of a data-in burst.

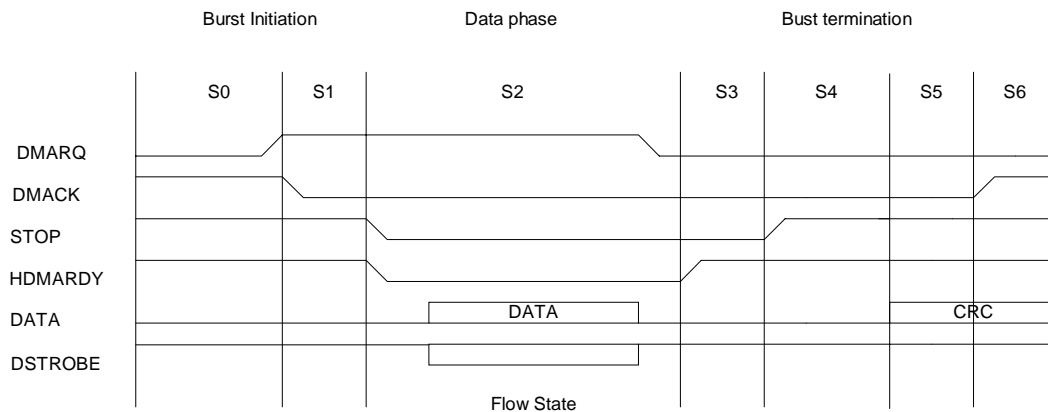


Figure 1 GPIF states used for UDMA transfer

Some drives will break up a UDMA burst request into multiple UDMA bursts. The GPIF waveforms must take this case into account by configuring the transaction count register for the total number of words requested. As shown below, each of the sub-UDMA bursts requires a CRC from the host. Since the CRC must be reset prior to each burst, the GPIF waveform must pass through the idle state in order to reset the CRC. While in idle, if the transaction count is not zero, GPIF will branch to state zero which is defined to wait for another UDMA request from the drive.

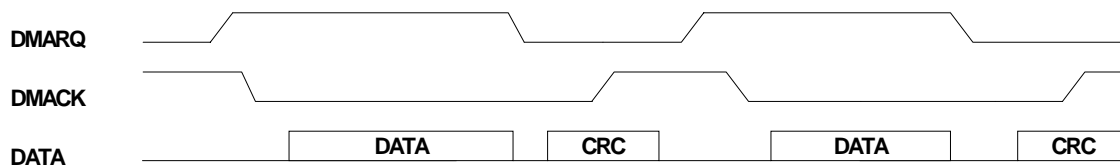


Figure 2 Example of drive forcing multiple UDMA transactions

During the data phase of a transaction the device receiving data must be able to hold-off the sending device in case of a full buffer condition, for example. UDMA allows for up to three additional words to be sent before the halt condition is recognized and data stops flowing. For a drive read, the host uses HDMARDY to throttle the drive. In this specific case, the host would assert HDMARDY when the IN end point buffer was almost full. PF is used with a value of the total buffer size – 3 words. In the write or out direction, if the end point buffer runs empty, the sending device can simply stop toggling the data strobe. If the drive needs to halt the data flow then it asserts DDMARDY.

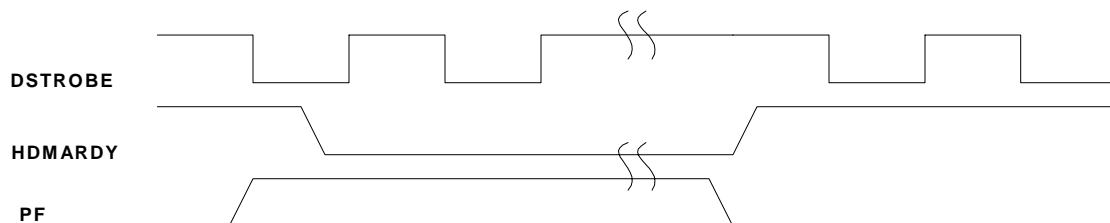


Figure 3 Host throttling drive during data phase of transfer.

All of the data throttling is managed by the flow state logic, specifically the FLOWEQ0CTL and FLOWEQ1CTL registers. More will be said about the application of these registers in the following example.

UDMA 66 Example

This next section of the document will illustrate how to implement a UDMA 66 interface using one of our development boards or reference designs. In general it is assumed that IFCONFIG will be configured for 30MHz, internal GPIF clock. For both the read and write UDMA transfers we will assume that the drive will terminate the transfer. Timing requirements for UDMA 66 that must be configured in the flow registers are as follows:

Strobe clock period 60ns Min
Data Setup/Hold 5ns Min

Since UDMA transactions are only in one direction for any given request, the documentation will describe the configuration for Data writes first and then for data reads.

UDMA 66 OUT , Data Writes to drive.

The following waveform defines the GPIF write states:

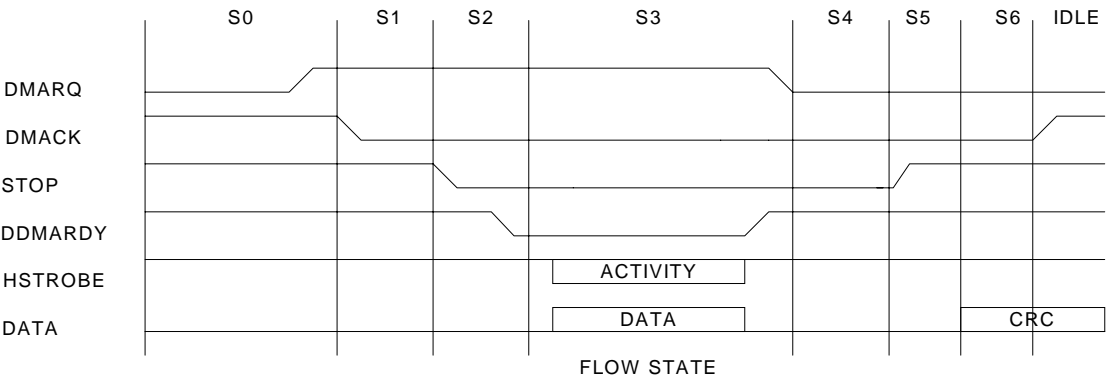


Figure 4 Drive terminated UDMA out GPIF waveform

In this state the host is the master generating the data and strobe. The strobe half period is defined in FLOWSTBHPERIOD. In our example we will use a value of x02h which will generate a half period of 2 x 16.67 ns = 33.33ns or 60 MBPS (Keep in mind that we transfer 16 bits with each edge of HSTROBE).

FLOWSTBHPERIOD								E6CD
Bits 7:0	masterstb_halfpd[7:0]							
Default	0	0	0	0	0	0	1	0
OUT 66	0	0	0	0	0	0	1	0
Access	RW	RW	RW	RW	RW	RW	RW	RW

To meet the data setup and hold requirements of UDMA we basically need to delay or hold the data with respect to the strobe edge by ½ clock cycle or 16.67nS. This is implemented by the GPIFHOLDAMOUNT register using x01. NOTE: GPIFHOLDAMOUNT will affect the data as well as the CRC data.

GPIFHOLDAMOUNT								E60C
Bits 7:0	0	0	0	0	0	0	hold_time[1:0]	
Default	0	0	0	0	0	0	0	0
OUT 66	0	0	0	0	0	0	0	1
Access	RW	RW	RW	RW	RW	RW	RW	RW

The following block diagram defines the physical interface between the host and drive. The assignment of the CTL and RDY signals are completely arbitrary, any CTL output or RDY input can be assigned as the master strobe (HSTROBE, DSTROBE). This example uses the hardware configuration found on our development system or reference design.

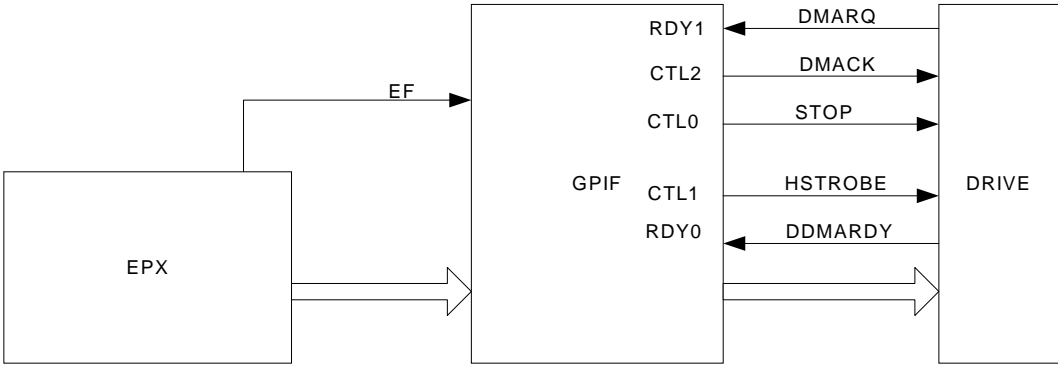


Figure 5 Hardware configuration for OUT (Data write to drive)

CTL1 is defined as the “Master Strobe” or HSTROBE. FLOWSTB is configured as show below:

FLOWSTB								E6CB
Bits 7:0	flow_slave_mode	RDY_async	CTL_togl_state	sustain_CTL	0	master_strobe[2:0]		
Default	0	0	1	0	0	0	0	0
OUT 66	0	don't care	0*	1	0	0	0	1
0	RW	RW	RW	RW	R	RW	RW	RW

- flow_slave_mode 0 GPIF is master of the bus transaction
- RDY_async X Don’t care
- CTL_togl_state 0 Defines FLOWLOGIC = 0 as the flow enable state
- sustain_CTL 1 Sustain the state of master strobe after the flow state
- master_strobe 001 Define CTL1 as the master strobe.

UDMA requires that both edges of the data strobe clock data. FLOWSTBEDGE is configured with 0x03 defining that both edges are active data strobes as follows :

FLOWSTBEDGE							E6CC	
Bits 7:0	0	0	0	0	0	0	falling_ edge	rising_ edge
Default	0	0	0	0	0	0	0	1
OUT 66	0	0	0	0	0	0	1	1
Access	R	R	R	R	R	R	RW	RW

All of the RDY inputs from the drive must be considered asynchronous signals. GPIFREADYCFG is configured for 0x40 as shown below:

GPIFREADYCFG								E6F3
Bits 7:0	INTRDY	SAS	tc_expire_ not_rdy5	0	0	0	0	0
Default	0	0	0	0	0	0	0	0
OUT 66	0	1	0	0	0	0	0	0
Access	R/W	R/W	R/W	R	R	R	R	R

While in the flow state, throttle conditions are managed by FLOWLOGIC, FLOWEQ0CTL, and FLOWEQ1CTL. For writes, two conditions will halt the flow of data, one if the host runs out of data EF= 1 or the device signals the host to stop DDMARDY = 1. FLOWLOGIC defines the conditions that resolve to FLOWLOGIC = 0 or 1 (Data is flowing or data is halted).

FLOWSTB[CTL_togl_state] = 0 defined FLOWLOGIC =0 as the state where data flow is enabled. FLOWLOGIC = 0 is the condition when DDMARDY = 0 and EF = 0. FLOWLOGIC will transition to 1 (Flow disabled) when DDMARDY = 1 OR EF = 1.

FLOWLOGIC								E6C7
Bits 7:0	flow_logic[1:0]		ready_A[2:0]			ready_B[2:0]		
Default	0	0	0	0	0	0	0	0
OUT 66	0	1	1	1	0	0	0	0
Access	RW	RW	RW	RW	RW	RW	RW	RW

flow_logic[1:0] 01 OR
ready_A[2:0] 110 Select EF. Note EP2GPIFFLGSEL must be set to use EF, 0x01
ready_B[2:0] 000 Select RDY0 which is DDMARDY in this example.

The following GPIF waveform will be used for the UDMA write:

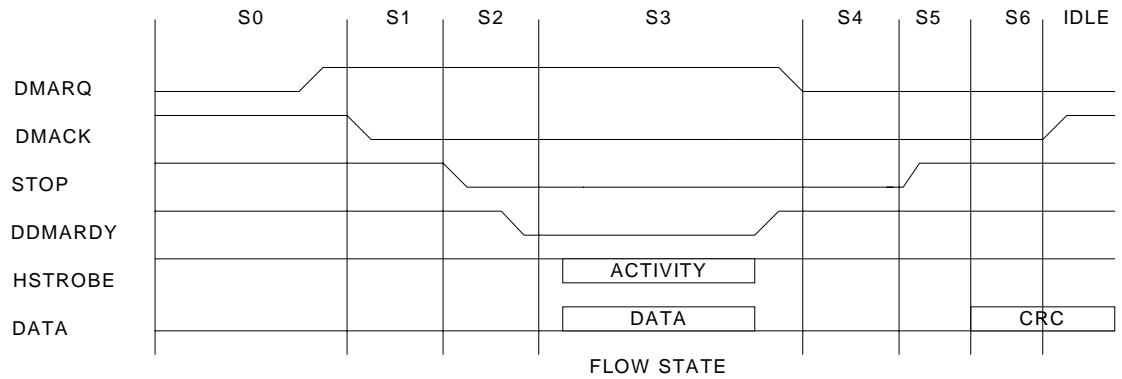


Figure 6 UDMA write GPIF waveform

UDMA writes do not require any change in state of the CTL signals, relative to how they were configured for the GPIF flow state S3, in response to FLOWLOGIC changes of state. Since CTL0 and CTL2 were defined to be 0 for state 3, the contents of FLOWEQ0CTL and FLOWEQ1CTL need to be 0x00. Note that CTL1 is don't care since it is defined as the “master strobe”.

FLOWEQ0CTL								E6C8
Bits 7:0	CTLOE3	CTLOE2	CTLOE1 / CTL5	CTLOE0 / CTL4	CTL3	CTL2	CTL1	CTL0
Default	0	0	0	0	0	0	0	0
OUT 66	0	0	0	0	0	0	0	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
OUT 66	0	0	0	0	0	0	0	0
Access	RW	RW	RW	RW	RW	RW	RW	RW

State three must be defined as the flow state for the write waveform and is done by configuring FLOWSTATE as follows:

FLOWSTATE								E6C6
Bits 7:0	flowstate_enable	0	0	0	0	flowstate[2:0]		
Default	0	0	0	0	0	0	0	0
OUT 66	1	0	0	0	0	0	1	1
Access	RW	R	R	R	R	RW	RW	RW

NOTE: Flow enable must only be set for GPIF waveforms that use the flow state, all other waveforms in order to work properly, must clear this bit.

See table 4 for GPIF tool waveform settings.

UDMA 66 in, Data Reads from drive

For drive reads, the following describes the interface (note the physical connections are the same but some of the signals have be redefined because of the change in direction).

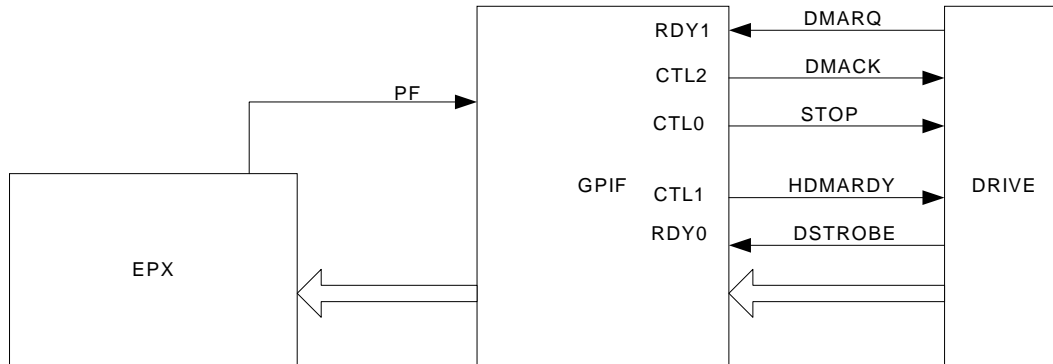


Figure 7 hardware configuration for IN (Data read from drive)

The following waveform defines the GPIF write states:

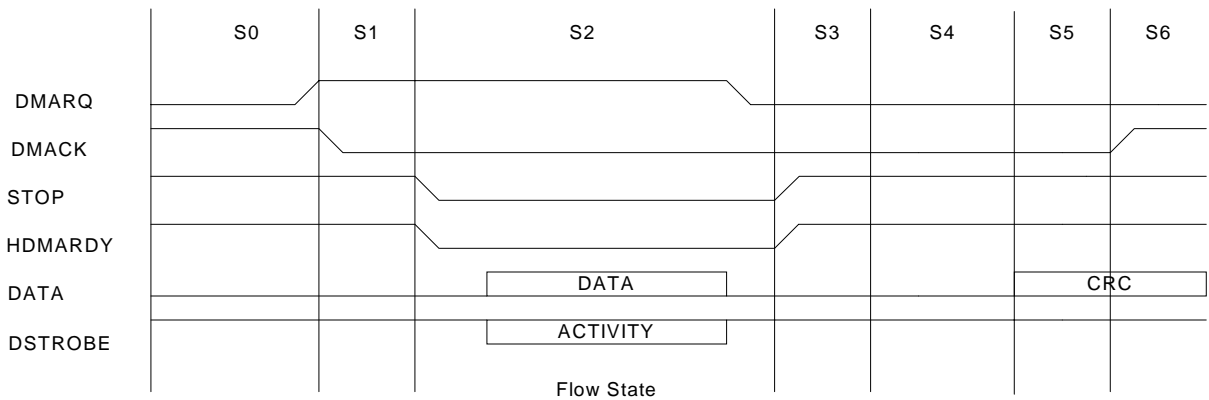


Figure 8 Drive terminated UDMA IN GPIF waveform

For Drive reads, the drive will be the Master generating the data strobe and data. Only the registers that are different from the OUT case will be discussed. Please refer to the register summary for a complete list of register setting for both cases.

RDY0 is defined as the “Master Strobe” or DSTROBE. FLOWSTB is configured as show below:

FLOWSTB						E6CB		
Bits 7:0	flow_slave_mode	RDY_async	CTL_togl_state	sustain_CTL	0	master_strobe[2:0]		
Default	0	0	1	0	0	0	0	0
IN 66	1	0	X	X	0	0	0	0
0	RW	RW	RW	RW	R	RW	RW	RW

flow_slave_mode 1 GPIF is slave of the bus transaction
RDY_async 0 RDY0 is not synchronous to IFCLK

CTL_togl_state	X	don't care
sustain_CTL	X	don't care
master_strobe	000	Define RDY0 as the master strobe.

While in the flow state, throttle conditions are managed by FLOWLOGIC, FLOWEQ0CTL, and FLOWEQ1CTL. For reads, the only condition that will halt the flow of data, other than the drive stopping DSTROBE, would be if the end point buffer fills up, PF= 1. HDMARDY (CTL1) is used to throttle the drive in this condition and is normally = 0 enabling the drive to transfer data. HDMARDY must be asserted to 1 in order to throttle the drive off. FLOWLOGIC = 0 when PF= 0, FLOWLOGIC = 1 when PF = 1.

FLOWLOGIC								E6C7
Bits 7:0	flow_logic[1:0]		ready_A[2:0]			ready_B[2:0]		
Default	0	0	0	0	0	0	0	0
IN 66	0	0	1	1	0	1	1	0
Access	RW	RW	RW	RW	RW	RW	RW	RW

flow_logic[1:0]	00	AND (could be OR since we are only looking at one signal)
ready_A[2:0]	110	Select PF. Note EPXGPIFFLGSEL must be set to use PF
ready_B[2:0]	110	Select RDY0 which is DDMARDY in this example.

NOTE: There are many ways to define the above logic, this is just one example.

Since the only signal that needs to change states in order to throttle the drive is HDMARDY (CTL1). In the FLOWEQ0CTL state CTL1 is 0, and one in the FLOWEQ1CTL state, 1.

FLOWEQ0CTL								E6C8
Bits 7:0	CTLOE3	CTLOE2	CTLOE1 / CTL5	CTLOE0 / CTL4	CTL3	CTL2	CTL1	CTL0
Default	0	0	0	0	0	0	0	0
IN 66	0	0	0	0	0	0	0	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
IN 66	0	0	0	0	0	0	1	0
Access	RW	RW	RW	RW	RW	RW	RW	RW

State two is defined as the flow state for the read waveform and is configuring as follows:

FLOWSTATE								E6C6
Bits 7:0	flowstate_enable	0	0	0	0	flowstate[2:0]		
Default	0	0	0	0	0	0	0	0
IN 66	1	0	0	0	0	0	1	0
Access	RW	R	R	R	R	RW	RW	RW

See table 5 for GPIF tool waveform settings.

		Reset		
E60C	GPIFHOLDAMOUNT	01	01	
E6C1	GPIFIDLECS	80	00	
E6C2	GPIFIDLECTL	FF	00	
E6C3	GPIFCTLCFG	00	00	
E6C6	FLOWSTATE	00	83	
E6C7	FLOWLOGIC	00	70	
E6C8	FLOWEQ0CTL	00	00	
E6C9	FLOWEQ1CTL	00	00	
E6CA	FLOWHOLDOFF	00	00	
E6CB	FLOWSTB	20	11	
E6CC	FLOWSTBEDGE	01	03	
E6CD	FLOWSTBHPERIOD	02	02	
E6F3	GPIFREADYCFG	00	40	

Table 2 Register summary for UDMA OUT, Drive writes

		Reset		
E60C	GPIFHOLDAMOUNT	01	01	
E6C1	GPIFIDLECS	80	00	
E6C2	GPIFIDLECTL	FF	00	
E6C3	GPIFCTLCFG	00	00	
E6C6	FLOWSTATE	00	82	
E6C7	FLOWLOGIC	00	36	
E6C8	FLOWEQ0CTL	00	00	
E6C9	FLOWEQ1CTL	00	02	
E6CA	FLOWHOLDOFF	00	00	
E6CB	FLOWSTB	20	80	
E6CC	FLOWSTBEDGE	01	03	
E6CD	FLOWSTBHPERIOD	02	02	
E6F3	GPIFREADYCFG	00	40	

Table 3 Register summary for UDMA IN, Drive read

Interval	0	1	2	3	4	5	6	Idle (7)
AddrMode	Same Val	Same Val	Same Val	Same Val	Same Val	Same Val	Same Val	
DataMode	NO Data	NO Data	Activate	Activate	Activate	Activate	Activate	
NextData	SameData	SameData	SameData	SameData	SameData	SameData	NextData	
Int Trig	No Int	No Int	No Int	No Int	No Int	No Int	No Int	
IF/Wait	IF	Wait 1	Wait 6	IF	Wait 1	Wait 1	Wait 2	
Term A	DMARQ			DMARQ				
LFunc	AND			AND				
Term B	DMARQ			DMARQ				
Branch1	Then 1			Then 3				
Branch0	Else 0			Else 4				
Re-Exec	No			No				
Sngl/CRC	Default	Default	Default	Default	Default	Default	Sngl/CRC	
STOP	1	1	0	0	0	1	1	1
HSTROBE	1	1	1	1	0	1	1	1
DMACK	1	0	0	0	0	0	0	1
CTL 3	1	1	1	1	1	1	1	1
CTL 4	1	1	1	1	1	1	1	1
CTL 5	1	1	1	1	1	1	1	1

Table 4 UDMA OUT, Drive Write, GPIF waveform

Interval	0	1	2	3	4	5	6	Idle (7)
AddrMode	Same Val	Same Val	Same Val	Same Val	Same Val	Same Val	Same Val	
DataMode	NO Data	NO Data	NO Data	NO Data	NO Data	Activate	Activate	
NextData	SameData	SameData	SameData	SameData	SameData	NextData	NextData	
Int Trig	No Int	No Int	No Int	No Int	No Int	No Int	No Int	
IF/Wait	IF	Wait 1	IF	Wait 1	Wait 1	Wait 2	Wait 1	
Term A	DMARQ		DMARQ					
LFunc	AND		AND					
Term B	DMARQ		DMARQ					
Branch1	Then 1		Then 2					
Branch0	Else 0		Else 3					
Re-Exec	No		No					
Sngl/CRC	Default	Default	Default	Default	Default	Sngl/CRC	Sngl/CRC	
STOP	1	1	0	1	1	1	1	1
HSTROBE	1	1	0	1	1	1	1	1
DMACK	1	0	0	0	0	0	1	1
CTL 3	1	1	1	1	1	1	1	1
CTL 4	1	1	1	1	1	1	1	1
CTL 5	1	1	1	1	1	1	1	1

Table 5 UDMA IN, Drive read, GPIF waveform