

IP-Datenblatt *axi_2d_mmvs*

Prof. Dr.-Ing. W. Gehrke, 11/2020

1. Übersicht

In vielen Anwendungsfällen ist es erforderlich, AXI-Stream-Daten aus dem externen Speicher zu lesen bzw. in dem externen Speicher abzulegen. Für den Zugriff auf den externen Speicher wird in der Regel der AXI-Standard verwendet. Für streamingbasierte Speicherzugriffe ist es wünschenswert, einen IP-Block einzusetzen, der Bereiche des externen Speichers lesen kann und die gelesenen Daten als AXI-Stream zur Verfügung stellt bzw. per AXI-Stream übergebene Daten im externen Speicher ablegt. Das IP *axi_2d_mmvs* stellt diese Funktionalität zur Verfügung.

Das IP kann sowohl über den IP-Konfigurationsdialog als auch mit Hilfe eines AXIL-Interfaces über Software konfiguriert werden. Bei entsprechender Konfiguration im Vivado IP-Integrator, ist es möglich, das IP auch ohne Software-Interaktion zu betreiben.

Das Modul *axi_2d_mmvs* ist vorrangig für die Ausführung von zweidimensionalen Speicherzugriffen mit dem Schwerpunkt Videostreaming konzipiert. Es kann jedoch auch für allgemeine (nicht videospezifische) Anwendungsfälle genutzt werden.

2. Funktionsweise

Das *axi_2d_mmvs* IP ist in die beiden Subblöcke *mm2vs* und *vs2mm* unterteilt. Das Modul *mm2vs* (main memory to video stream) dient dem Auslesen des externen Speichers und stellt die Daten als AXI-Stream zur Verfügung. Der Subblock *vs2mm* (video stream to main memory) schreibt Daten, die als AXI-Stream zur Verfügung gestellt werden, in den Speicher.

Die beiden Subblöcke arbeiten unabhängig voneinander und erlauben einen gleichzeitigen Schreib- und Lese-Zugriff.

AXI-spezifische Eigenschaften wie die 4kB-Boundary werden vom IP unterstützt. Darüber hinaus wird das Pipelining von Lesezugriffen unterstützt.

2.1. Speicheradressierung

Die Adressierung des Speichers erfolgt für beide Subblöcke gleich und wird in diesem Abschnitt gemeinsam für die Module *mm2vs* und *vs2mm* beschrieben. Hierbei wird die für Videodaten übliche Terminologie (z.B. "Zeile" oder "Bild") verwendet. Dennoch kann das IP auch für beliebige Streaminganwendungen, z.B. für Audiodaten, genutzt werden.

Für die Adressierung des Speichers werden die folgenden Angaben (im Konfigurationsdialog bzw. über AXIL-Register) benötigt:

- **Startadresse des Speicherbereichs**

Die Startadresse verweist auf das erste Byte des verwendeten Speicherbereichs. Die Größe des genutzten Bereichs ergibt sich aus den nachfolgend beschriebenen Parametern.

Die Adresse muss "aligned" sein: Bei AXIS-Wortbreite 32 bit muss die Startadresse durch 4 teilbar sein, bei einer Wortbreite von 64 bit, ist eine durch 8 teilbare Startadresse erforderlich.

- **Anzahl der Bytes in einer Zeile**

Dieser Parameter gibt die Anzahl der Bytes in einer Zeile an. Der Parameter muss wie die Adresse "aligned" sein.

- **Stride**
Stride bezeichnet den Adressoffset (in Bytes) zwischen dem Beginn zweier aufeinanderfolgenden Zeilen. Liegen die Zeilen direkt ohne Lücke hintereinander, ist dieser Wert also identisch zu der Anzahl der Bytes pro Zeile.
Der Parameter muss wie die Adresse "aligned" sein.
- **Anzahl der Zeilen eines Bildes**
Mit Hilfe dieses Parameters wird die Anzahl der Zeilen eines Bildes angegeben.
- **Anzahl der Bildspeicher**
Im einfachsten Fall nutzt das IP einen Speicherbereich dessen Größe der Größe eines einzelnen Bildes entspricht. Auf diese Weise ist der Speicherbedarf zwar minimal, aber dies bedeutet auch, dass mit dem Beginn eines Bildes neue Daten in dem gleichen Speicherbereich geschrieben (mm2vs: gelesen) werden, der für das vorangegangene Bild verwendet wurde. Die Folge hiervon kann ein Datenverlust sein, wenn die Daten des vorangegangenen Bildes noch nicht von anderen Komponenten des Systems konsumiert (mm2vs: produziert) worden sind. Diese Problematik kann durch die Verwendung von mehreren Bildspeichern entschärft werden.
Das IP bietet die Möglichkeit, **zusätzlich** zu nutzenden Bildspeicher anzugeben. **Diese liegen im externen Speicher ohne Lücke hintereinander und werden von dem IP zyklisch adressiert.**
Interrupts werden unabhängig von gewählten der Anzahl der Bildspeicher in jedem Bild generiert.

2.2. AXIS-Interfaces

Das IP besitzt einen AXIS-Master- (MM2VS) und einen AXI-Slave-Port (VS2MM).

Neben den üblichen AXIS-Signalen wie *DATA*, *VALID* und *READY*, werden vom Master-Port auch die Signale *LAST* und *USER* entsprechend dem AXIS-Video-Standard der Firma Xilinx erzeugt: *USER*=1 kennzeichnet das erste Element eines Bildes, während mit *LAST*=1 das letzte Element eine Zeile markiert wird.

Für das *USER*-Signal (= Beginn eines Bildes, Start of Frame, *SOF*) stehen zwei Optionen zur Verfügung: Einerseits kann das *USER*-Signal ignoriert werden oder es wird zu Beginn der Adressierung eines neuen Bildes auf *USER*=1 gewartet, so dass die Inhalte der Bildspeicher immer mit dem Beginn eines neuen Bildes synchronisiert sind – was insbesondere bei Videostreaming von Bedeutung ist. Wird die Synchronisierung gewählt, werden alle eingehenden Daten verworfen, die zwischen dem Ende eines Bildes und der *SOF*-Signalisierung über den AXIS-Slave-Port geliefert werden.

Das *LAST*-Signal wird vom Slave-Port immer ignoriert. Die Adressierung innerhalb eines Bildes erfolgt nur auf Grundlage der konfigurierten Adressparameter.

2.3. Integrierte FIFO-Speicher

Das IP besitzt integrierte FIFO-Speicher zu Entkopplung der Speicherzugriffe und der AXIS-Kanäle. Auf diese Weise können am AXI-Slave-Port Daten entgegengenommen werden, selbst wenn der Speicherzugriff, zum Beispiel aufgrund des Zugriffs anderer Speicherklienten, verzögert ist. Ebenso kann der AXI-Master-Port im Fall eines verzögerten Speicherzugriffs Daten zu Verfügung stellen.

Darüber hinaus können die FIFOs als Clock-Bridge fungieren und ermöglichen es, die AXIS-Ports mit einem anderen Takt als den AXI-Speicher-Port zu betreiben.

Die Größe der FIFO-Speicher wird über den IP-Konfigurationsdialog gewählt. Zu beachten ist, dass hierbei die Adresswortbreite des FIFO-Speichers ausgewählt wird. Das

FIFO enthält also $2^{\text{Adresswortbreite}}$ Einträge mit der gewählten Datenwortbreite (32 bzw. 64 bit).

2.4. Synchronisation: Interrupts und Finished Pulse

Für die Synchronisation mit Softwarekomponenten des Systems können von jedem der beiden Subblöcke Interrupts erzeugt werden. Die Interrupts werden mit dem Ende einer (wählbaren) Bildzeile erzeugt.

Die Interruptleitungen werden mit dem Ende der gewählten Zeile auf 1 gesetzt und müssen durch Löschen der zugehörigen Bits im Interrupt-Status-Register zurückgesetzt werden¹.

Die Auswahl der Zeile, an dessen Ende ein Interrupt ausgelöst wird, wird durch einen konfigurierbaren Zeilenoffset festgelegt. Ist der Offset 0, wird der Interrupt am Ende eines Bildes ausgelöst. Ist der Offset größer als Null, erfolgt die Auslösung des Interrupts um die angegebene Zeilenanzahl früher. Beispiel: Bei einem Offsetwert von 1 wird der Interrupt mit Ende der vorletzten Zeile ausgelöst.

Darüber hinaus kann die HW-HW-Synchronisation über zwei "Impulsausgänge" erfolgen. Die Ausgänge *MM2VS_FINISHED_PULSE* bzw. *VS2MM_FINISHED_PULSE* erzeugen mit dem Interrupt-Ereignis einen High-Impuls mit der Länge eines Taktzyklusses des AXI-Speicher-Interfaces.

2.5. AXIL-Interface

Das AXIL-Interface dient der Konfiguration des IPs zur Laufzeit. Neben Steuerregistern stehen auch Statusregister zur Verfügung, mit denen unter anderem die aktuelle Zeile, das aktuelle Bild und andere Statusinformationen ausgelesen werden können.

Wird das IP mit unterschiedlichen Taktsignalen betrieben, ist zu beachten, dass bei der Implementierung des AXIL-Interfaces auf Maßnahmen zur Taktsynchronisierung verzichtet wurde. Dies wird als sinnvoll angesehen, da das IP, abgesehen von der Bestätigung von Interrupts, nie im laufenden Betrieb (Run-Bit = 1) umkonfiguriert wird bzw. werden sollte.

Sollen Änderungen an der Konfiguration zur Laufzeit vorgenommen werden, muss zunächst der entsprechende Subblock angehalten und auf die Beendigung aller laufenden und gepipelineten Speicherzugriffe gewartet werden. Erst danach darf die Konfiguration des Subblocks modifiziert werden.

Im "Last Frame Start Register" wird die bei einem Bildstart verwendete Adresse abgelegt. Mit Hilfe dieser Information kann ermittelt werden, welcher der angelegten Bildspeicher aktuell von dem IP verwendet wird bzw. welche(r) Bildspeicher unbenutzt sind und z.B. durch Softwarekomponenten geschrieben bzw. gelesen werden können.

¹ Das ist nicht "interruptfest" und wird in einer der nachfolgenden Versionen des IPs modifiziert bzw. erweitert.

3. IP-Konfiguration

Der Konfigurationsdialog des IPs besitzt die drei Reiter *General*, *MM2VS* und *VS2MM*.

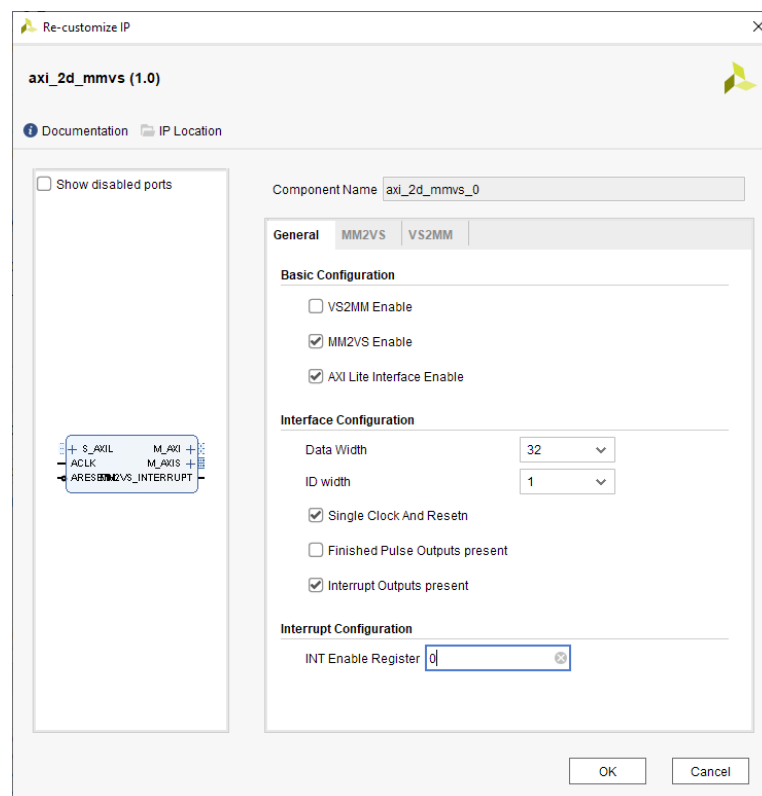
Unter dem Reiter *General* kann ausgewählt werden

- ob die Subblöcke *MM2VS* bzw. *VS2MM* zur Verfügung gestellt werden sollen
- ob ein AXIL-Interface benötigt wird
- welche Datenwortbreite für die AXI(S)-Interfaces verwendet wird
- ob ein gemeinsame oder getrennte Takt- und Resetsignale verwendet werden
- ob Interrupt- und Finished-Pulse-Ausgänge benötigt werden
- welchen Wert das Interrupt-Enable-Register nach dem Reset besitzen soll

Unter den Reitern *MM2VS* und *VS2MM* können die Adressparameter, die FIFO-Größe, die maximal zu verwendende Burstlänge, der Wert der Signale *ARCACHE* und *AWCACHE*² sowie der Interruptzeilenoffset eingestellt werden. Im Fall des *MM2VS*-Moduls kann zusätzlich die Anzahl der gepipelineten Bursts ausgewählt werden.

Im Abschnitt *MM2VS Control* bzw. *VS2MM Control* kann der Resetwert des "Run-Bits" gewählt werden. Ist dieses Bit zu 1 gewählt, startet das IP unmittelbar nach einem Reset. Diese Funktion ist für Anwendungsfälle gedacht, in der keine Software-Interaktion mit dem IP erfolgen soll und zum Beispiel kein AXIL-Interface verwendet wird.

Die Einstellung "Synchronisation with Start of Frame (SOF)" ist nur für das Modul *VS2MM* relevant und dient der Synchronisation des Speicherschreibzugriffs mit dem Start eines Bildes (*USER=1*).



² vgl. AXI-Spezifikation

Re-customize IP

axi_2d_mmvs (1.0)

Documentation IP Location

☐ Show disabled ports

Component Name: axi_2d_mmvs_0

General **MM2VS** VS2MM

MM2VS Memory Buffer Configuration

MM2VS: Startaddress: 0x38000000

MM2VS: Horizontal Size [Bytes]: 1024 [16 - 65550]

MM2VS: Stride [Bytes]: 1024 [0 - 65528]

MM2VS: Vertical Size [Lines]: 192 [1 - 65534]

MM2VS: Number of additional (Frame) Buffers: 1 [0 - 14]

MM2VS Control

MM2VS: Run: 1

MM2VS: Synchronisation with Start of Frame (SOF): 0

MM2VS FIFO Configuration

MM2VS: FIFO Address Width: 9

MM2VS Memory Interface Configuration

MM2VS: Maximum Additional Pipelined Bursts: 3

MM2VS: Maximum Burst Length: 16

MM2VS: AxCACHE Setting: 0 [0 - 15]

MM2VS Interrupt Configuration

MM2VS INT Line: 0 [0 - 65533]

OK Cancel

Re-customize IP

axi_2d_mmvs (1.0)

Documentation IP Location

☐ Show disabled ports

Component Name: axi_2d_mmvs_0

General MM2VS **VS2MM**

VS2MM Memory Buffer Configuration

VS2MM: Startaddress: 0x38000000

VS2MM: Horizontal Size [Bytes]: 1024 [16 - 65550]

VS2MM: Stride [Bytes]: 1024 [0 - 65528]

VS2MM: Vertical Size [Lines]: 1024 [1 - 65534]

VS2MM: Number of additional (Frame) Buffers: 1 [0 - 14]

VS2MM Control

VS2MM: Run: 0

VS2MM: Synchronisation with Start of Frame (SOF): 0

VS2MM FIFO Configuration

VS2MM: FIFO Address Width: 9

MM2VS Memory Interface Configuration

VS2MM: Maximum Burst Length: 16

VS2MM: AxCACHE Setting: 0 [0 - 15]

VS2MM Interrupt Configuration

VS2MM INT Line: 0 [0 - 65534]

OK Cancel

4. Registerbelegung

Dieser Abschnitt fasst die Belegung der Register des IPs tabellarisch zusammen.

Register	Offset	Bits								
Interrupt Register										
Interrupt Enable	0x00	31:2					1		0	
		reserved					1: VS2MM INT Enable		1: MM2VS INT Enable	
Interrupt Status	0x04	31:2					1		0	
		reserved					VS2MM INT Status		MM2VS INT Status	
MM2VS Register										
MM2VS Control	0x08	31:16	15:12	11:10	9	8	7:4	3:2	1	0
		reserved	ARCACHE	resvd	freeze	resvd	# add. frame buffers	resvd	Sync w SOF	Run (1) / Stop (0)
MM2VS Start Address	0x0C	31:0								
		Buffer start address								
MM2VS Horizontal Bytes	0x10	31:16					15:0			
		reserved					Number of Bytes per Line			
MM2VS Stride	0x14	31:24					23:0			
		reserved					Stride [Bytes]			
MM2VS Lines	0x18	31:16					15:0			
		reserved					Number of lines			
MM2VS INT Line Offset	0x1C	31:16					15:0			
		reserved					Line offset for INT generation			
MM2VS Frame Num (read only)	0x20	31:16					15:0			
		reserved					Current Frame Number			
MM2VS Line Num (read only)	0x24	31:16					15:0			
		reserved					Current Line Number			
MM2VS Last Frame Start Address (read only)	0x28	31:0								
		Last Frame Start Address used by MM2VS								
VS2MM Register										
VS2MM Control	0x2C	31:16	15:12	11:10	9	8	7:4	3:2	1	0
		reserved	AWCACHE	resvd	freeze	resvd	# add. frame buffers	resvd	Sync w SOF	Run (1) / Stop (0)
VS2MM Startaddress	0x30	31:0								
		Buffer start address								
VS2MM Horizontal Bytes	0x34	31:16					15:0			
		reserved					Number of Bytes per Line			
VS2MM Stride	0x38	31:24					23:0			
		reserved					Stride [Bytes]			
VS2MM Lines	0x3C	31:16					15:0			
		reserved					Number of lines			
VS2MM INT Line Offset	0x40	31:16					15:0			
		reserved					Line offset for INT generation			
VS2MM Frame Num (read only)	0x44	31:16					15:0			
		reserved					Current Frame Number			
VS2MM Line Num (read only)	0x48	31:16					15:0			
		reserved					Current Line Number			
VS2MM Last Frame Start Address (read only)	0x4C	31:0								
		Last Frame Start Address used by VS2MM								