

Projektiranje programabilnih SoC platformi

Predavanje II - FER, 2022.

Projektiranje programabilnih SoC platformi

- Današnje predavanje
- Video senzor ov7670
 - Osnove video senzora
 - Video sabirniva - ZV port
 - Kontrolna sabirnica SCCB
- AMBA sabirnički sustav
 - AMBA Ukratko
 - AXI
 - AXI memory mapped
 - AXI Lite
 - AXI Stream
 - Podjela borbenih kompleta

Video Senzor

- Dvije vrste - tehnološki različite
 - CCD (charge coupled device)
 - CMOS (complementary metal oxide semiconductor)
- Svaka tehnologija ima svoje prednosti i mane u ovisnosti o primjeni. Niti jedan nema superiornu prednost pred drugim iako često možete naći da proizvođači tvrde suprotno.

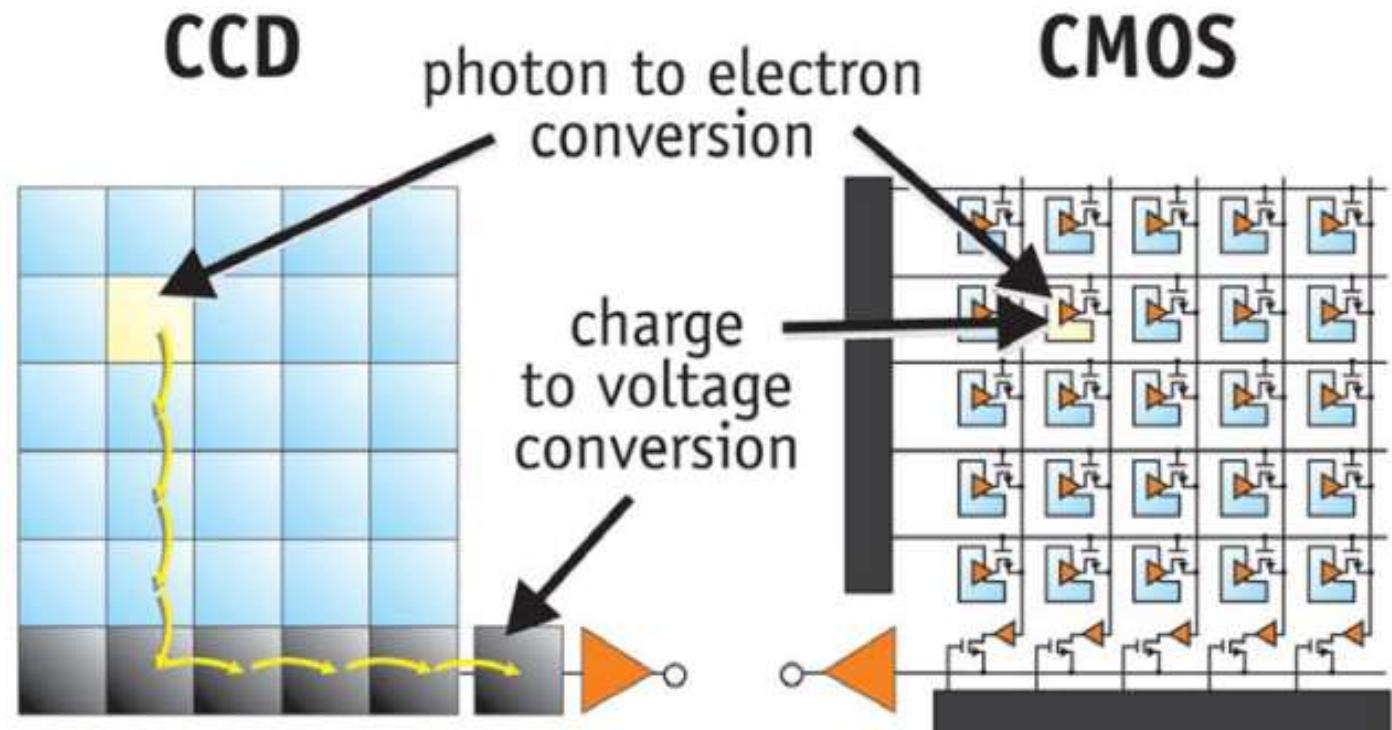
Video Senzor

- CCD i CMOS izumljeni su kasnih '60 i početkom '70 (osnivač DALSA Dr. Savvas Chamberlain).
- Zbog svoje strukture i jednostavnije izvedbe (čitaj cijene ☺) CCD senzori su postali dominantni.
- Napretkom tehnologije '90 godina ponovo se pojavljuje interes za CMOS senzorima

Video Senzor

- **CCD senzor**

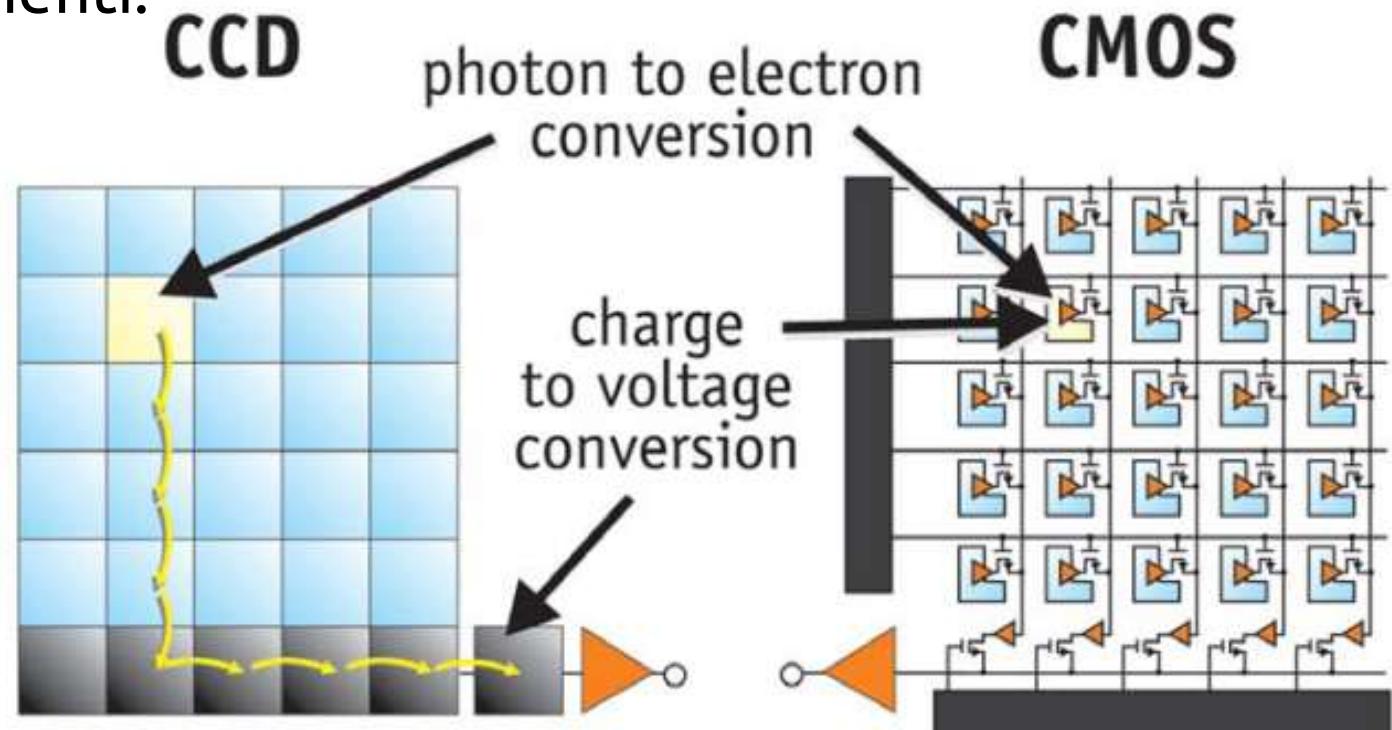
- naboj svakog piksela prenosi se vrlo kratkim putovima i kroz mali broj čvorova prije nego se pretvori u naponsku razinu i pošalje izvan čipa kao analogni signal (koristi se samo jedan ili mali broj pretvarača). Svi pikseli dohvaćaju se istovremeno i uniformno se obrađuju (osnovni uvjet kvalitetne slike).



Video Senzor

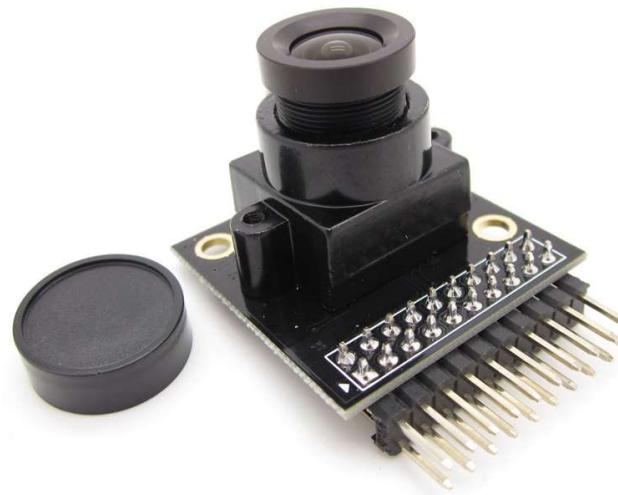
- **CMOS senzor:**

- Svaki piksel ima svoj zasebni sklop za pretvorbu naboja u napon. Senzori obično posjeduju i pojačala, filtre šuma, i digitalni sklop sve na jednom čipu.
- Sve navedeno povećava fleksibilnost dizajna.
- Pošto svaki piksel ima svoj pretvarač narušava se uniformnost konverzije, ali zato čip može biti izrađen da treba mali broj vanjskih komponenti.



Video Senzor

- OmniVision's OV7670 SINGLE-CHIP CMOS VGA COLOR DIGITAL CAMERA



- 640x480 – VGA format
- 24 pina
- Podržan format – YUV 4:2:2, GRB 4:2:2, RGB Raw Data 565/555
- 8 video data: ITU-601, ITU-656, ZV port
- Automatska ekspozicija/gain/kontrola bijele boje (WB)
- Operacije nad slikom – svjetloća, kontrast, gamma, saturation, oštrina, uzimanje dijela slike, i.t.d.
- Vanjska i unutarnja sinkronizacija
- Frame exposure/line exposure option (za foto aparate)

OV7670

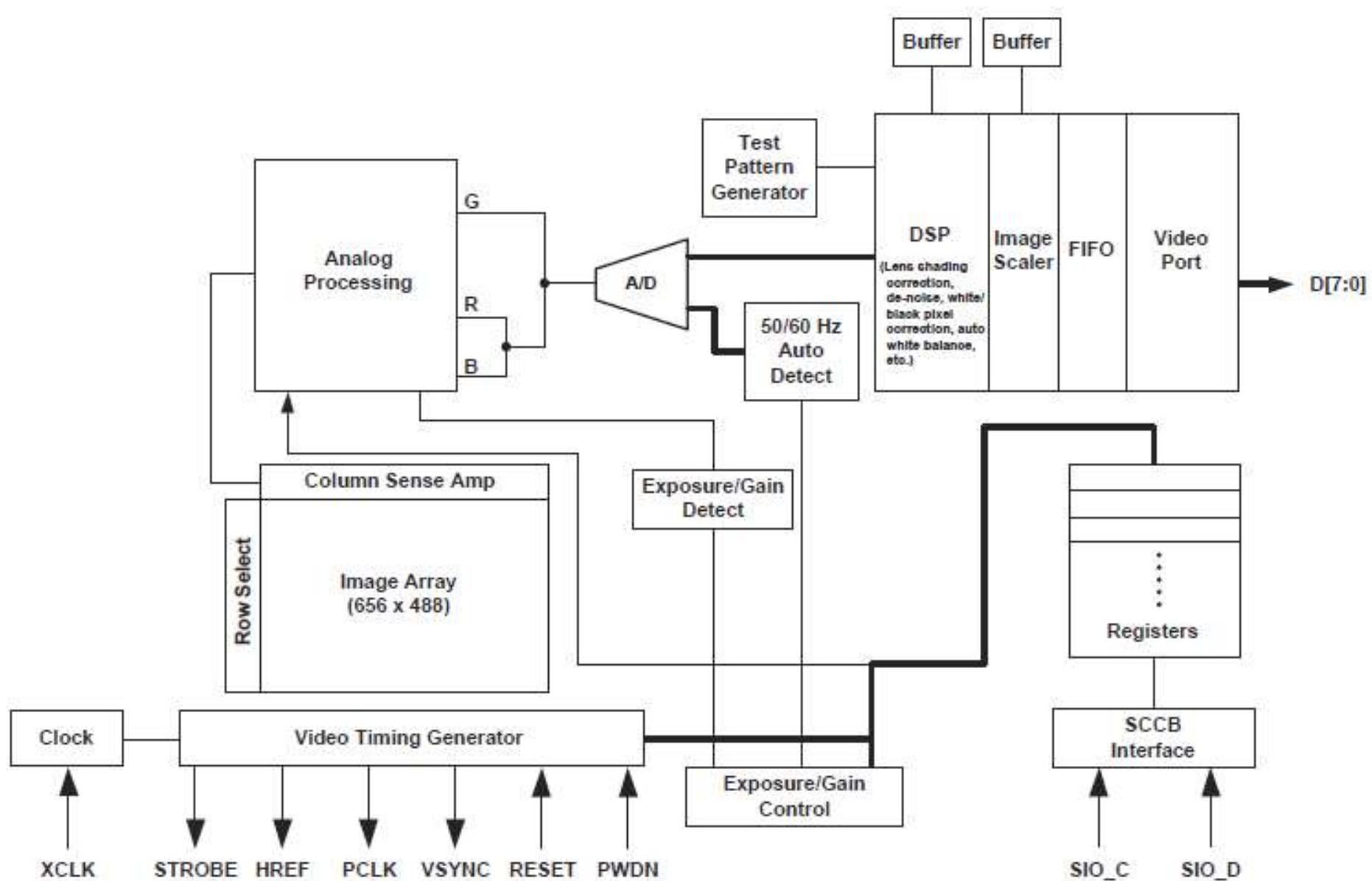
- 3.3V Volt operation, low power dissipation
 - < 80 mW active power
 - < 20 uA in power-save mode
- SCCB kontrolno sučelje (400 kb/s):
 - Color saturation, brightness, contrast, white balance, exposure time, gain

OV7670

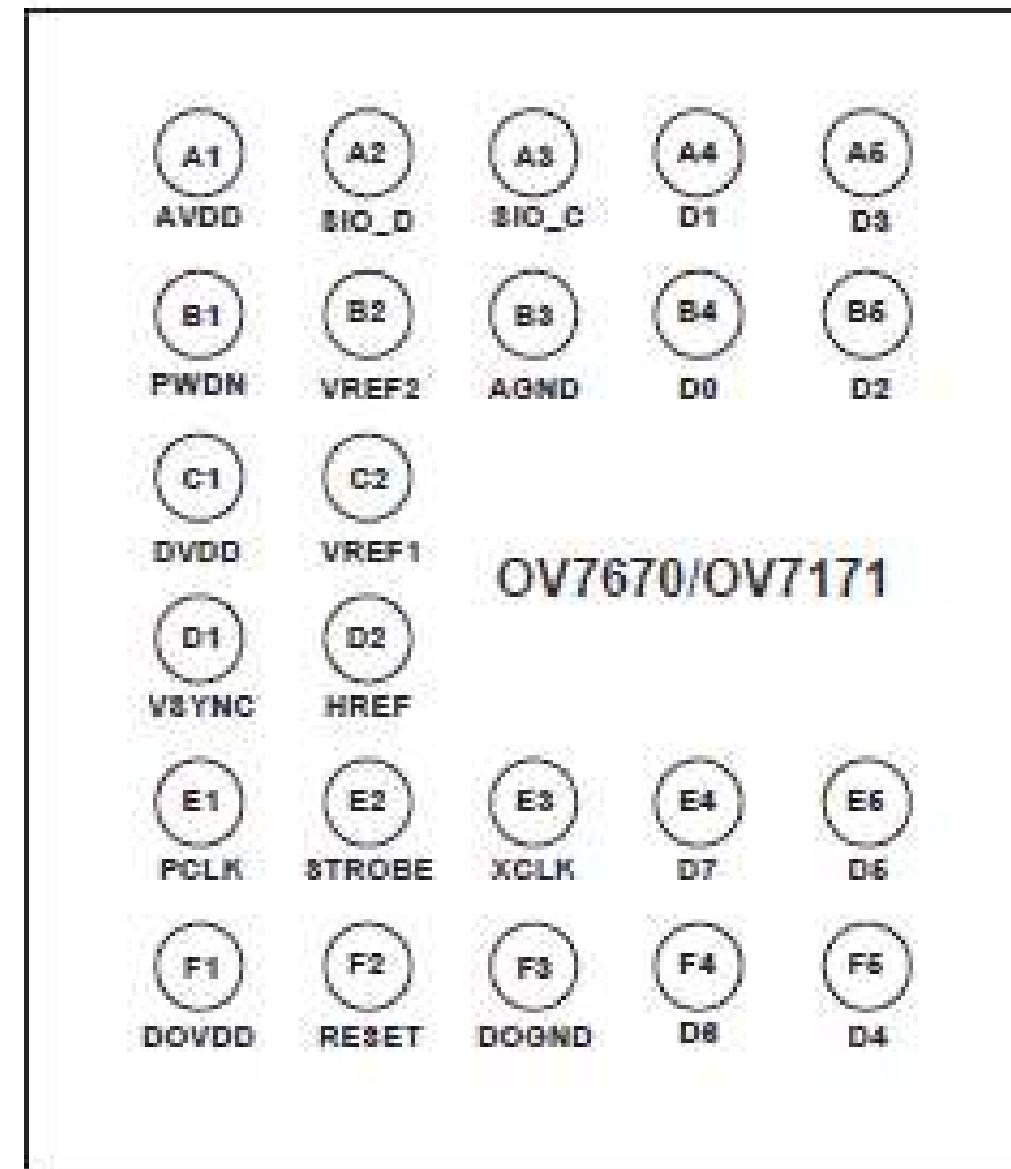
- Postoji i crno-bijela verzija OV7171
- CMOS Video senzor
- Ukupan broj pikslea 656×488
- Maksimalno do 60 slika po sekundi

- Senzor je bio predviđena za sljedeće funkcije:
 - Video konferencije
 - Video telefonija
 - Foto aparati na mobilnim telefonima
 - PC Multimedia
 - i.t.d.

OV7670



OV7670



- **Image Sensors**

- 200-megapixel
- 100-megapixel
- 64-megapixel
- 40-megapixel
- 20-megapixel
- 16-megapixel
- 14-megapixel
- 13-megapixel
- 12-megapixel
- 10-megapixel
- i.t.d.

- Dva komunikacijska sučelja
 - Kontrolno sučelje SCCB (I2C)
 - Podatkovno sučelje: ZV port, (ITU-601, ITU-656)

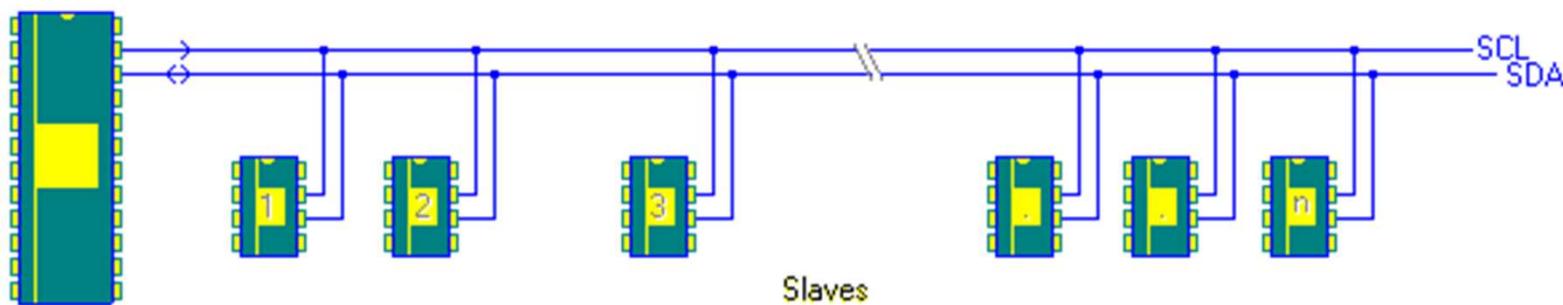
SCCB - Serial Camera Control Bus

- OmniVision komunikacijski protokol koji koriste u svim svojim senzorima
- Omogućava komunikaciju jednog master uređaja i više slave uređaja
- Vrlo sličan I2C protokolu

- I2C - Inter-Integrated Circuit
- Početkom '80ih Philips Semiconductors razvio je dvosmjernu "2-wire" komunikacijsku sabirnicu.
- Osnovna namjena joj je bila da omogući jednostavnu komunikaciju između procesora i ostalih komponenti unutar televizora.
- Philips Labs u Eindhoven (Nizozemskoj)
- Danas I2C široko primijenjen i u drugim uređajima.

- Fizički je izvedena sa dvije žice (spojna puta)
 - SDA – Serial DAta
 - SCL – Serial CLock
- Obije linije su dvosmjerne
- Svaki uređaj spojen na sabirnicu mora imati svoju jedinstvenu adresu na toj sabirnici
- Svaki od uređaja može slati ili primati podatke

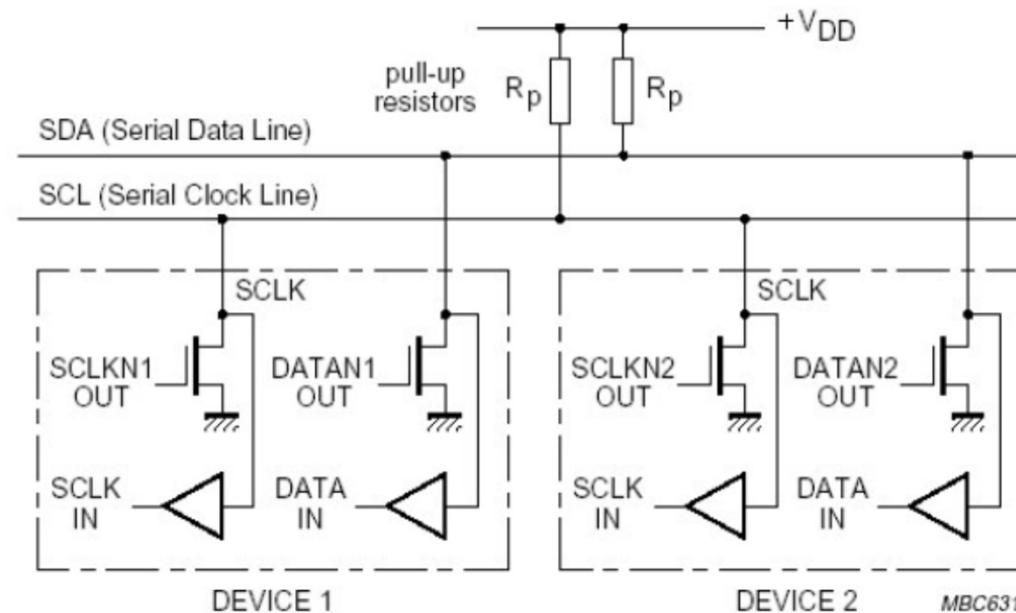
- Svojstva sabirnice
 - Više uređaja može započeti komunikaciju
 - Uređaj koji započinje komunikaciju naziva se master na sabirnici
 - Sukladno tome svi ostali u tom trenutku su slave uređaji
 - Master na sabirnici je obično procesor
 - Sabirnica može imati više master uređaja

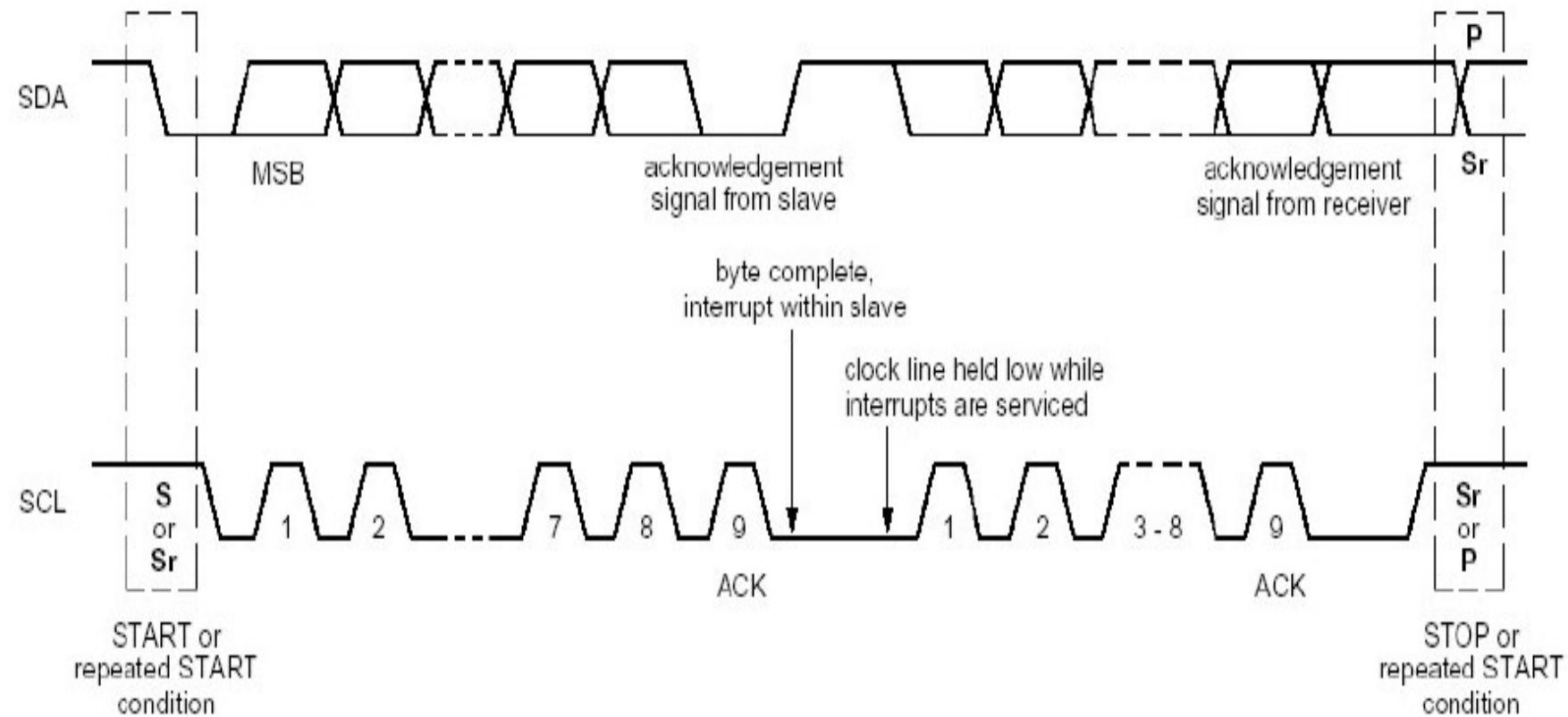


- Brzina komunikacije

- 100 kbps (standardni način rada)
- 400 kbps(brzi način rada)
- 3.4 Mbps (vrlo-brzi načina rada)
- Broj uređaja na sabirnici je limitiran s maksimalnim kapacitetom od 400 pF

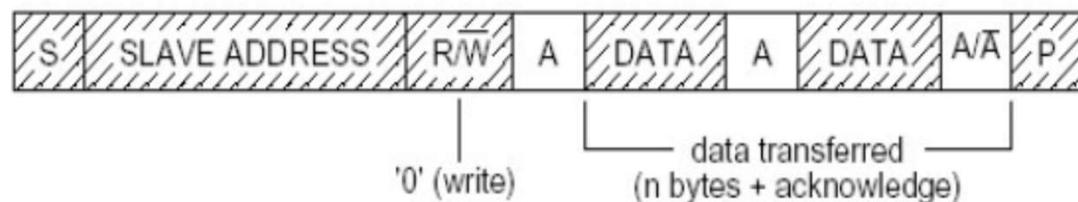
- Koristi sabirnicu Spojeno „I“
- Sabirnica je slobodna kada su SDA i SCL u visokom.
- Koriste se pull-up otpornici





- Okvir komunikacije:

master-transmitter



from master to slave

from slave to master

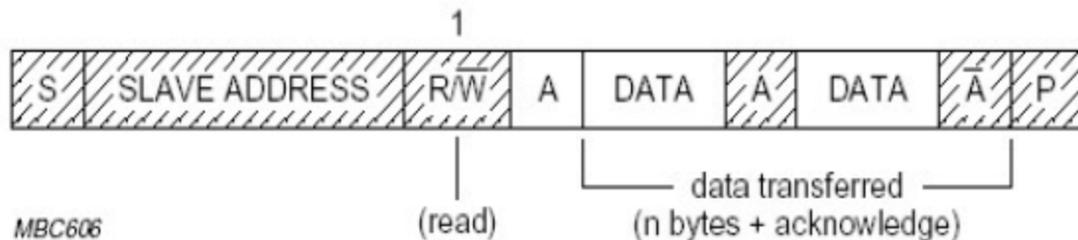
A = acknowledge (SDA LOW)

\bar{A} = not acknowledge (SDA HIGH)

S = START condition

P = STOP condition

master-receiver (since second byte)



MBC606

- Podržava brzinu prijenosa do 400 kbps
- Može imati samo jedan master:
 - nema arbitraže na podatkovnoj sabirnici
 - Nema sinkronizacije SCL (signala vremenskog vođenja)

- Podržava brzinu prijenosa do 400 kbps
- Može imati samo jedan master:
 - nema arbitraže na podatkovnoj sabirnici
 - Nema sinkronizacije SCL (signala vremenskog vođenja)
- Osim osnovnog adresiranje preko kojeg se odabire senzor s kojim se komunicira SCCB podržava i pod adresiranje internih registara unutar video senzora
- Tijekom slanja podataka (pisanja) nakon adrese senzora u drugom bajtu se šalje adresa регистра kojem se želi pristupiti
- Treći bajt koji se šalje zapisuje se u odabrani registar
- U koliko se nastavi dalje slati bajtovi podatci se zapisuju u sljedeće registre koji slijede

- Promjera smjera komunikacije kod SCCB i I2C nije moguća tijekom jedne komunikacije.
- Zato za čitanje s neke adrese prvo moramo imati jedan ciklju spisanja a nakon toga možemo čitat registe.
- Odnosno čitanje se vrši sa zadnje adresiranog registra u procesu pisanja
- Moramo koristiti jedan prazan ciklus pisanja u kojem ne upisujemo podatak u registar, odnosno šaljemo samo dva bajta – adresu senzora i adresu registra

- Nama bitni registri unutar senzora:

Register	Address	Default	Description
CLKRC	0x11	0x80	<p><i>Bit[6]:</i> 0: Apply prescaler on input clock 1: Use external clock directly</p> <p><i>Bit[0-5]: Clock prescaler</i> $F(\text{internal clock}) = F(\text{input clock}) / (\text{Bit}[0-5] + 1)$ Range [0 0000] to [1 1111]</p>
DBLV	0x6B	0x0A	<p><i>Bit[7-6]: PLL control</i></p> <p>00: Bypass PLL 01: Input clock x4 10: Input clock x6 11: Input clock x8</p> <p><i>Bit[4]: Regulator control</i></p> <p>0: Enable internal regulator 1: Bypass internal regulator</p>

- Nama bitni registri

70	SCALING_XSC	4A	RW	<p>Bit[7]: Test_pattern[0] - works with test_pattern[1] test_pattern (SCALING_XSC[7], SCALING_YSC[7]):</p> <p>00: No test output</p> <p>01: Shifting "1"</p> <p>10: 8-bar color bar</p> <p>11: Fade to gray color bar</p> <p>Bit[6:0]: Horizontal scale factor</p>
71	SCALING_YSC	35	RW	<p>Bit[7]: Test_pattern[1] - works with test_pattern[0] test_pattern (SCALING_XSC[7], SCALING_YSC[7]):</p> <p>00: No test output</p> <p>01: Shifting "1"</p> <p>10: 8-bar color bar</p> <p>11: Fade to gray color bar</p> <p>Bit[6:0]: Vertical scale factor</p>

OV7670

```
/*
 * Clock scale: 3 = 15fps
 *      2 = 20fps
 *      1 = 30fps
 */
[REG_CLKRC, 0x1],    /* OV: clock scale (30 fps) */
[REG_TSLB, 0x04],    /* OV */
[REG_COM7, 0],       /* VGA */

[REG_HSTART, 0x13],  { REG_HSTOP, 0x01 },
[REG_HREF, 0xb6],    { REG_VSTART, 0x02 },
[REG_VSTOP, 0x7a],   { REG_VREF, 0xa },
[REG_COM3, 0],       { REG_COM14, 0 },
/* Mystery scaling numbers */
[REG_SCALING_XSC, 0x3a],
[REG_SCALING_YSC, 0x35],
{0x72, 0x11},        {0x73, 0x0 },
{0xa2, 0x02},        {REG_COM10, 0x0 },
/* Gamma curve values */
{0x7a, 0x20},        {0x7b, 0x10 },
{0x7c, 0x1e},        {0x7d, 0x35 },
{0x7e, 0x5a},        {0x7f, 0x69 },
{0x80, 0x76},        {0x81, 0x80 },
{0x82, 0x88},        {0x83, 0x8f },
{0x84, 0x96},        {0x85, 0xa3 },
{0x86, 0xaf},        {0x87, 0xc4 },
{0x88, 0xd7},        {0x89, 0xe8 },
/* AGC and AEC parameters. Note we start by disabling those features,
then turn them only after tweaking the values. */
[REG_COMB, COM8_FASTAEC | COMB_AECSTEP | COMB_BFILT],
[REG_GAIN, 0],        { REG_AECH, 0 },
[REG_COM4, 0x40],    /* magic reserved bit */
[REG_COM9, 0x18],    /* 4x gain + magic red bit */
[REG_BDSOMAX, 0x05], { REG_BD60MAX, 0x07 },
[REG_AEW, 0x95],     { REG_AEB, 0x33 },
[REG_VPT, 0xe3],     { REG_HAECC1, 0x78 },
[REG_HAECC2, 0x68],   {0xa1, 0x03}, /* magic */
[REG_HAECC3, 0xd8],   { REG_HAECC4, 0xd8 },
[REG_HAECC5, 0xf0],   { REG_HAECC6, 0x90 },
[REG_HAECC7, 0x94],
[REG_COMB, COM8_FASTAEC | COM8_AECSTEP | COM8_BFILT | COM8_AGC | COM8_AEC ],
/* Almost all of these are magic "reserved" values. */
[REG_COM5, 0x61],    { REG_COM6, 0x4b },
{0x16, 0x02},         { REG_MVFP, 0x07 },
{0x21, 0x02},         {0x22, 0x91 },
{0x29, 0x07},         {0x33, 0x0b },
{0x35, 0x0b},         {0x37, 0x1d },
{0x38, 0x71},         {0x39, 0x2a },
[REG_COM12, 0x78],   {0x4d, 0x40 },
{0x4e, 0x20},         { REG_GFIX, 0 },
{0x6b, 0x4a},         {0x74, 0x10 },
{0x8d, 0x4f},         {0x8e, 0 },
{0x8f, 0},            {0x90, 0 },
{0x91, 0},            {0x96, 0 },
[0x9a, 0],             {0xb0, 0x84 },
{0xb1, 0x0c},          {0xb2, 0x0e },
{0xb3, 0x82},          {0xb8, 0x0a },
/* More reserved magic, some of which tweaks white balance */
{0x43, 0x0a},          {0x44, 0x0f0 },
{0x45, 0x34},          {0x46, 0x58 },
{0x47, 0x28},          {0x48, 0x3a },
{0x59, 0x88},          {0x5a, 0x88 },
{0x5b, 0x44},          {0x5c, 0x67 },
{0x5d, 0x49},          {0x5e, 0x0e },
{0x6c, 0x0a},          {0x6d, 0x55 },
{0x6e, 0x11},          {0x6f, 0x9f }, /* "9e for advance AWB" */
{0x6a, 0x40},          { REG_BLUE, 0x40 },
[REG_RED, 0x60 ],
[REG_COMB,
COMB_FASTAEC | COMB_AECSTEP | COMB_BFILT | COMB_AGC | COMB_AEC | COMB_AWB ],
/* Matrix coefficients */
{0x4f, 0x80},          {0x50, 0x80 },
{0x51, 0},              {0x52, 0x22 },
{0x53, 0x5e},          {0x54, 0x80 },
{0x58, 0x9e },
[REG_COM16, COM16_AWBGAIN], { REG_EDGE, 0 },
{0x75, 0x05},          {0x76, 0xe1 },
{0x4c, 0},              {0x77, 0x01 },
[REG_COM13, 0xc3],     {0x4b, 0x09 },
{0xc9, 0x60},          { REG_COM16, 0x38 },
{0x56, 0x40 },
{0x34, 0x11},          { REG_COM11, COM11_EXP | COM11_HZAUTO },
{0x4a, 0x88},          {0x96, 0 },
{0x97, 0x30},          {0x98, 0x20 },
{0x99, 0x30},          {0x9a, 0x84 },
{0x9b, 0x29},          {0x9c, 0x03 },
{0x9d, 0x4c},          {0x9e, 0x3f },
{0x78, 0x04 },
/* Extra-weird stuff. Some sort of multiplexor register */
{0x79, 0x01},          {0xc8, 0xf0 },
{0x79, 0x0f},          {0xc8, 0x00 },
{0x79, 0x10},          {0xc8, 0x7e },
{0x79, 0x0a},          {0xc8, 0x80 },
{0x79, 0x0b},          {0xc8, 0x01 },
{0x79, 0x0c},          {0xc8, 0x0f },
{0x79, 0x0d},          {0xc8, 0x20 },
{0x79, 0x09},          {0xc8, 0x80 },
{0x79, 0x02},          {0xc8, 0xc0 },
{0x79, 0x03},          {0xc8, 0x40 },
{0x79, 0x05},          {0xc8, 0x30 },
{0x79, 0x26 },
[0xff, 0xff ], /* END MARKER */
```

- <https://github.com/torvalds/linux/blob/master/drivers/media/i2c/ov7670.c>

XILINX ZYNQ

- Xilinx ZYNQ 7020 - I2C
 - PS
 - dva I2C sklopa
 - Može ih se spojiti na predefinirane pinove ili proslijediti u PL sklop
 - PL
 - IP core za I2C
 - Možemo dodati proizvoljan broj sklopova

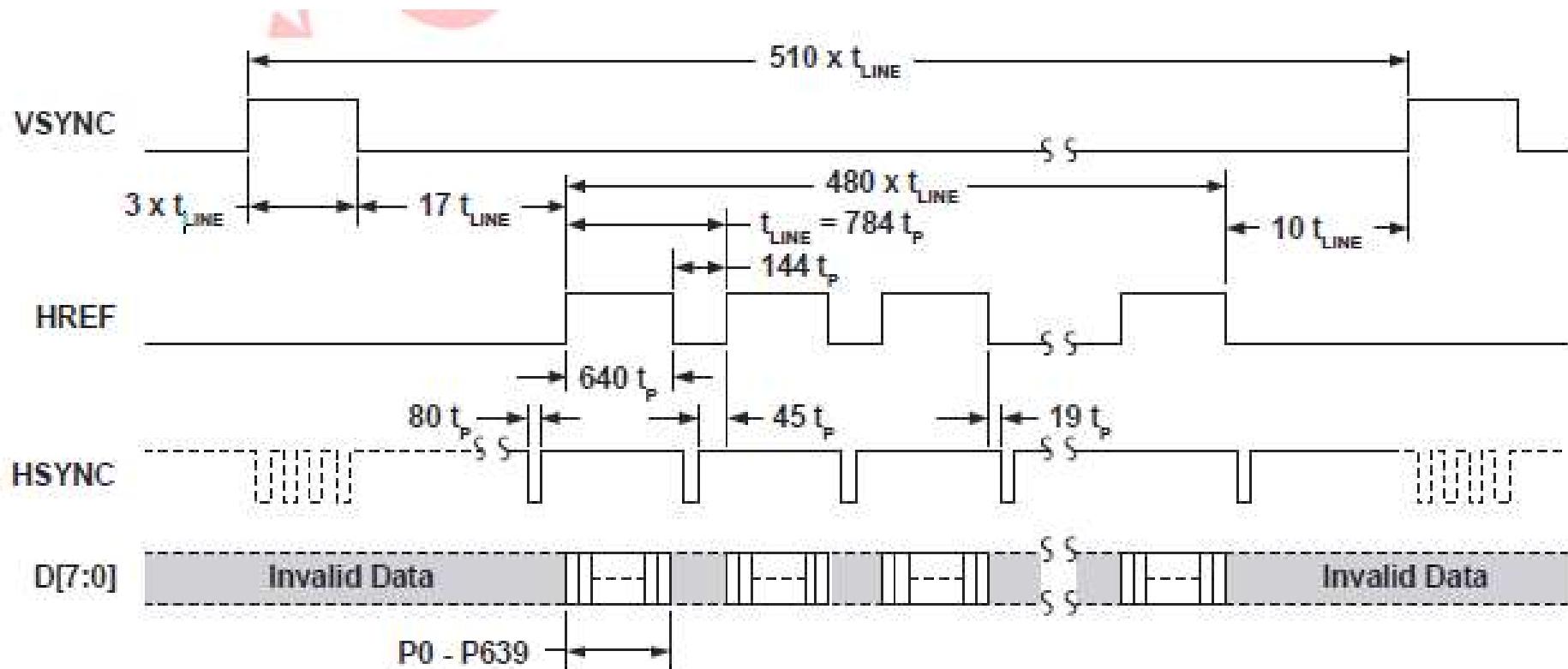
- Video sučelje
 - Video senzor podržava
 - 8 bitno digitalno sučelje
 - Video format CCIR601, CCIR656, ZV port
 - Format podataka – YUV 4:2:2, RGB 4:2:2, RGB row dana 565/555

- Pravi naziv BT.656 predložen od strane ITU (International Telecommunication Union) stoga se često naziva ITU656 i CCIR656
- Definira jednostavni video protokol za prijenos ne komprimiranog digitalnog video signala bilo PAL ili NTSC (522 ili 625 linija)
- Standar je nastao na BT.601(CCIR601) koji definira prijenos podataka u formatu 4:2:2 interlaced u YUV (YCbCr)
- Standard definira prijenos 8 i 10 bitnih podataka serijski ili paralelno.
- Koristi se za prijenos podataka u televizorima između čipova

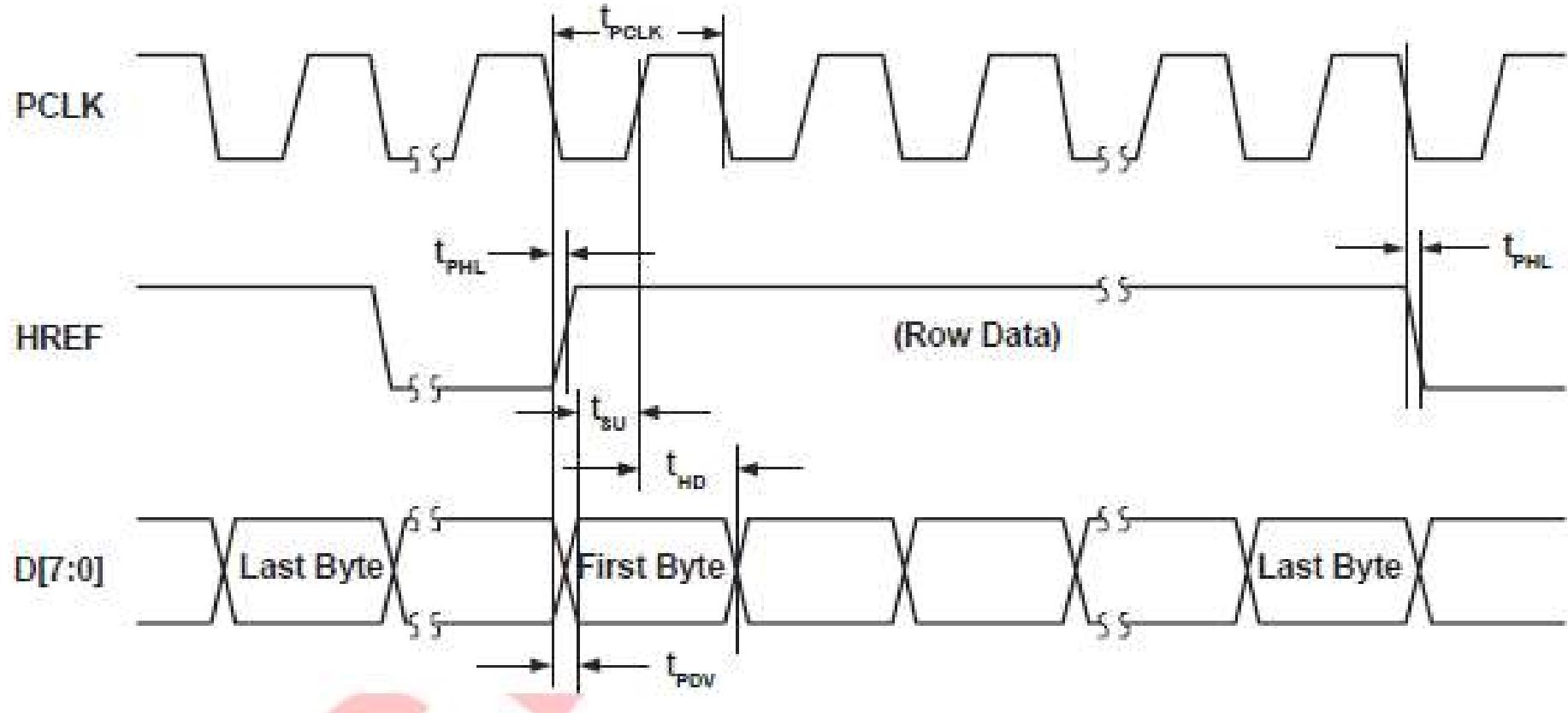
- ZV Port je dio PC Card standarda koji definira PCMCIA i PC Express kartice
- ZV (Zoomed Video) je protokol koji služi za povezivanje PC Card (PCMCIA) i host sistema (računala) koji omogućava direktni pristup video memoriji i/ili VGA upravljačkom sklopu. Podatci se prenose bez upotrebe međuspremnika odvojenom sabirnicom tako da ne opterećuju računalo.
- Pogodan je za jednostavno i jeftino povezivanje video uređaja koji zahtijevaju brzi prijenos podataka za aplikacije kao što su MPEG dekoderi za filmove i igrice, TV tuners, live video input and video capture.

- Više detalja u PC Card Standard
- Dodano u standard od verzije PC Card Standard 5.04 Update
- Trenutna verzija (koja je ujedno i konačna)
- PC Card Standard 8.0
- PC Card je napušten
- Postoji i Express Card koji podržava sve što i PC Card
- Fizički nisu kompatibilni

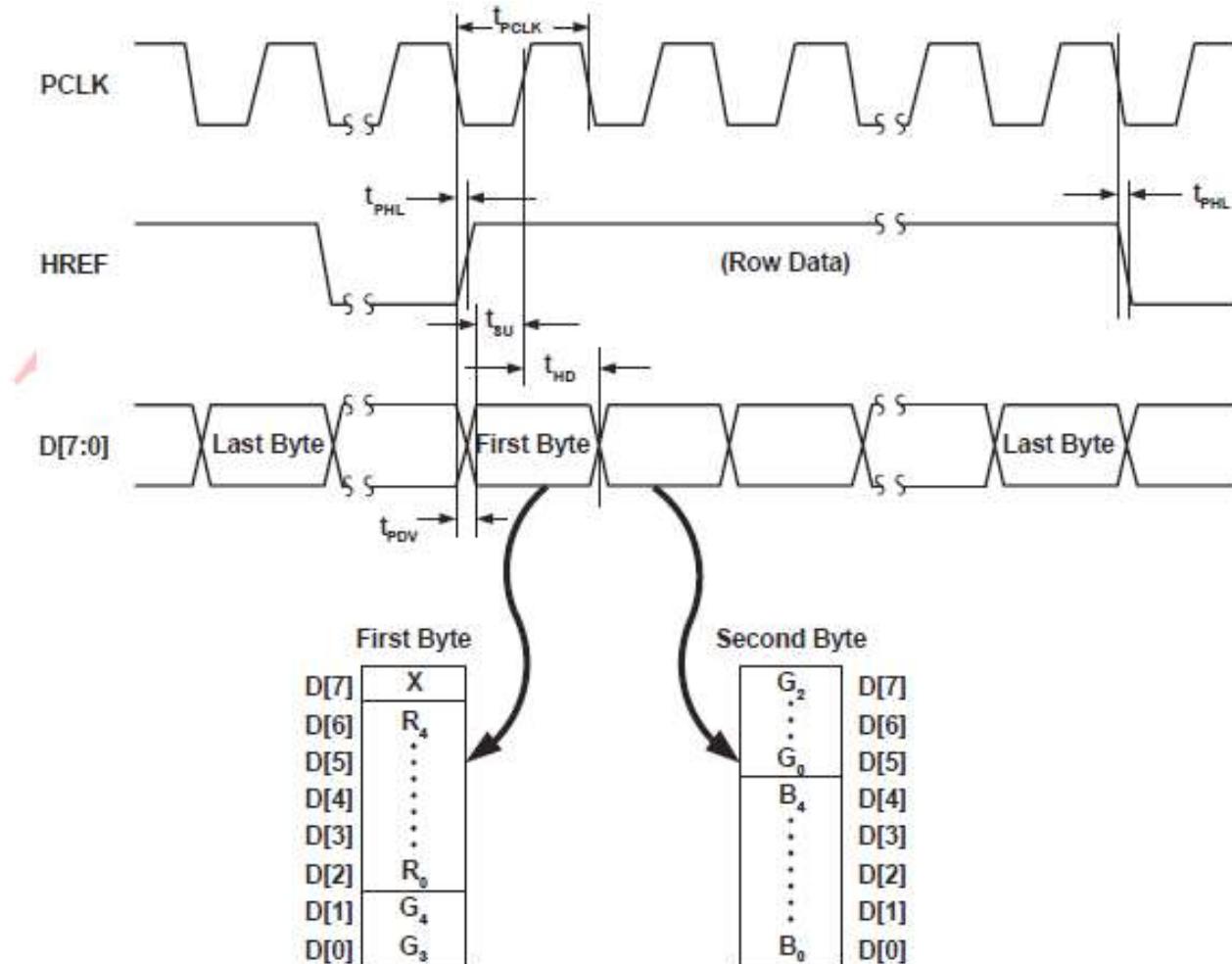
- Signalni
 - VSYNC
 - Postavljanjem u visoko dojavljuje se početak okvira slike. Generira se jednom za svaku sliku
 - HREF
 - Postavlja se visoko i ostaje u visokom kada se šalju jedan red slike po završetku slanja jednog reda postavlja se u nisko
 - PCLK
 - Na padajući ili rastući brid signala podatak na sabirnici je valjani
 - Opcija rastući ili padajući brid može se podesiti preko SCCB-a
 - D[7:0]
 - Osam bitna podatkovna sabirnica. Služi za prijenos podataka
 - RST
 - reset senzora
 - PWD
 - Postavljanje senzora u način rada štednje energije
 - PCLK
 - Izlazni clock po kome je sinkronizirana komunikacija
 - XCLK
 - Ulagani signal vremenskog vođenja bez kojeg senzor ne radi (10-28Mhz)



OV 7670



OV7670 – RGB565



OV7670

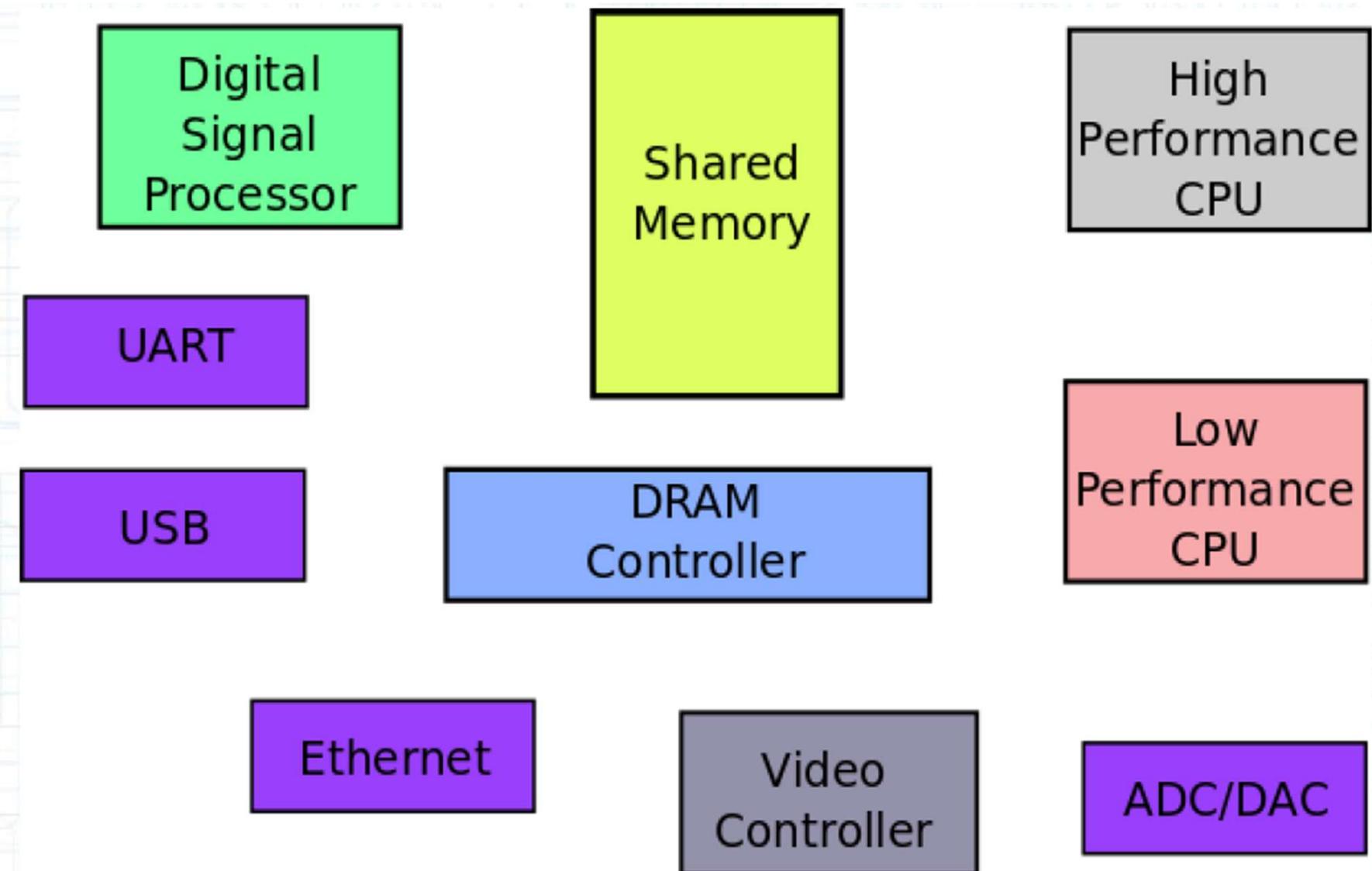
	Byte 0	Byte 1	Byte 2	Byte 3
Format	D7 D6 D5 D4 D3 D2 D1 D0	D7 D6 D5 D4 D3 D2 D1 D0	D7 D6 D5 D4 D3 D2 D1 D0	D7 D6 D5 D4 D3 D2 D1 D0
RGB 565	R4 R3 R2 R1 R0 G5 G4 G3	G2 G1 G0 B4 B3 B2 B1 B0	R4 R3 R2 R1 R0 G5 G4 G3	G2 G1 G0 B4 B3 B2 B1 B0
RGB 555	X R4 R3 R2 R1 R0 G4 G3	G2 G1 G0 B4 B3 B2 B1 B0	0 R4 R3 R2 R1 R0 G4 G3	G2 G1 G0 B4 B3 B2 B1 B0
RGB 444 xRGB	X X X X R3 R2 R1 R0	G3 G2 G1 G0 B3 B2 B1 B0	X X X X R3 R2 R1 R0	G3 G2 G1 G0 B3 B2 B1 B0
RGB 444 RGBx	R3 R2 R1 R0 G3 G2 G1 G0	B3 B2 B1 B0 X X X X	R3 R2 R1 R0 G3 G2 G1 G0	B3 B2 B1 B0 X X X X
YUV	Y U Y V			
	Y V Y U			
	U Y V Y			
	V Y U Y			

- Vaš zadatak
 - Koristimo jedno od dva I2C sučelje koje se nalazi na ZYNQ-u kontrolirati rad senzora
 - Korištenjem predefiniranih IP-ova spojiti senzor na VDMA kontroler
 - Definirati pretvorbu ZV Port sučelja u AXI_stream. Koristimo gotovi VHDL kod
- Napisati program za konfiguraciju kamere preko I2C porta
- Dohvatiti sliku i poslati je preko ethernet sučelja
- Na računalu dohvatiti sliku i prikazati je na ekranu
- Bez problema se može dobiti video stream.

AMBA Sabirnički sustav

Sabirnički sustavi

- Komponente od kojih se sastoje SoC sustavi

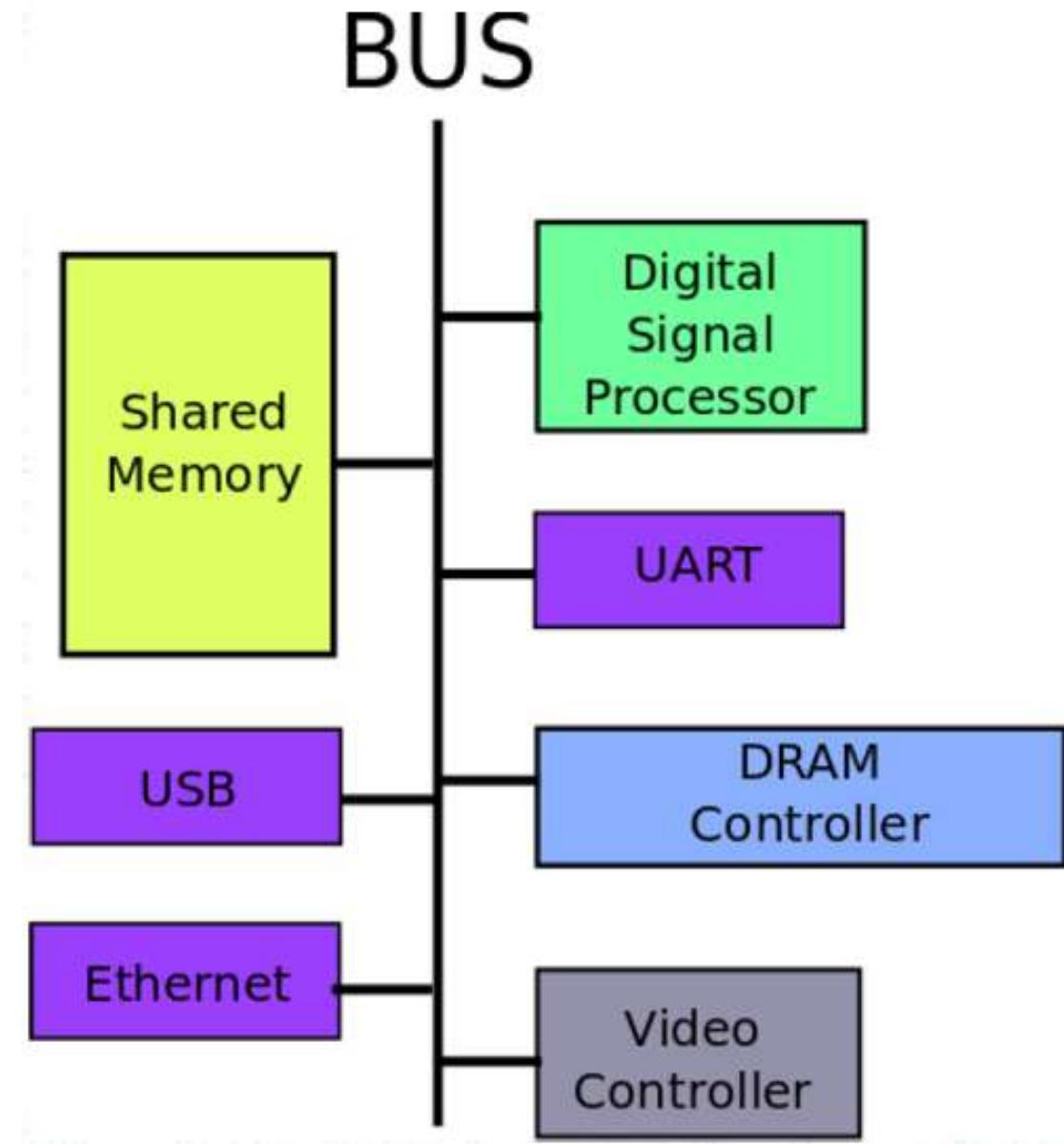


Sabirnički sustavi

- Zahtijeci pred sustave za povezivanje (sabirničke sustave)
 - Standardizacija
 - komponente sustava komuniciraju po definiranom standardu
 - Mogu komunicirati međusobno jedna s drugom
 - Sustav mora biti održiv:
 - Sustav mora biti jednostavan za održavanje, nadogradnju i otklanjanje pogreški
 - Ponovno korištenje (re-use)
 - Komponente sustava možemo opet koristiti

Sabirnički sustavi

- Standardni način na koji komuniciraju komponente na sabirnici



Sabirnički sustavi

- **Advanced Microcontroller Bus Architecture (AMBA):**
an open-standard, on-chip interconnect specification for the connection and management of functional blocks in system-on-a-chip (SoC) designs.
First version introduced by ARM in 1996.

- AMBA has evolved since 1996
 - currently in its 5th generation.
 - APB (Advanced Peripheral Bus) and ASB (Advanced System Bus) were the first of AMBA bus protocols.
 - AMBA 2 version in 1999 introduced AHB (Advanced High-Performance Bus).
 - AMBA 3 was introduced in 2003 that included AXI (Advanced eXtensible Interface).
 - AMBA 4 introduced ACE (AXI Coherency Extensions) protocol in 2010 and AMBA 5 introduced CHI (Coherent Hub Interface) in 2013.

Key AMBA Specifications

AMBA generation:

AMBA 2

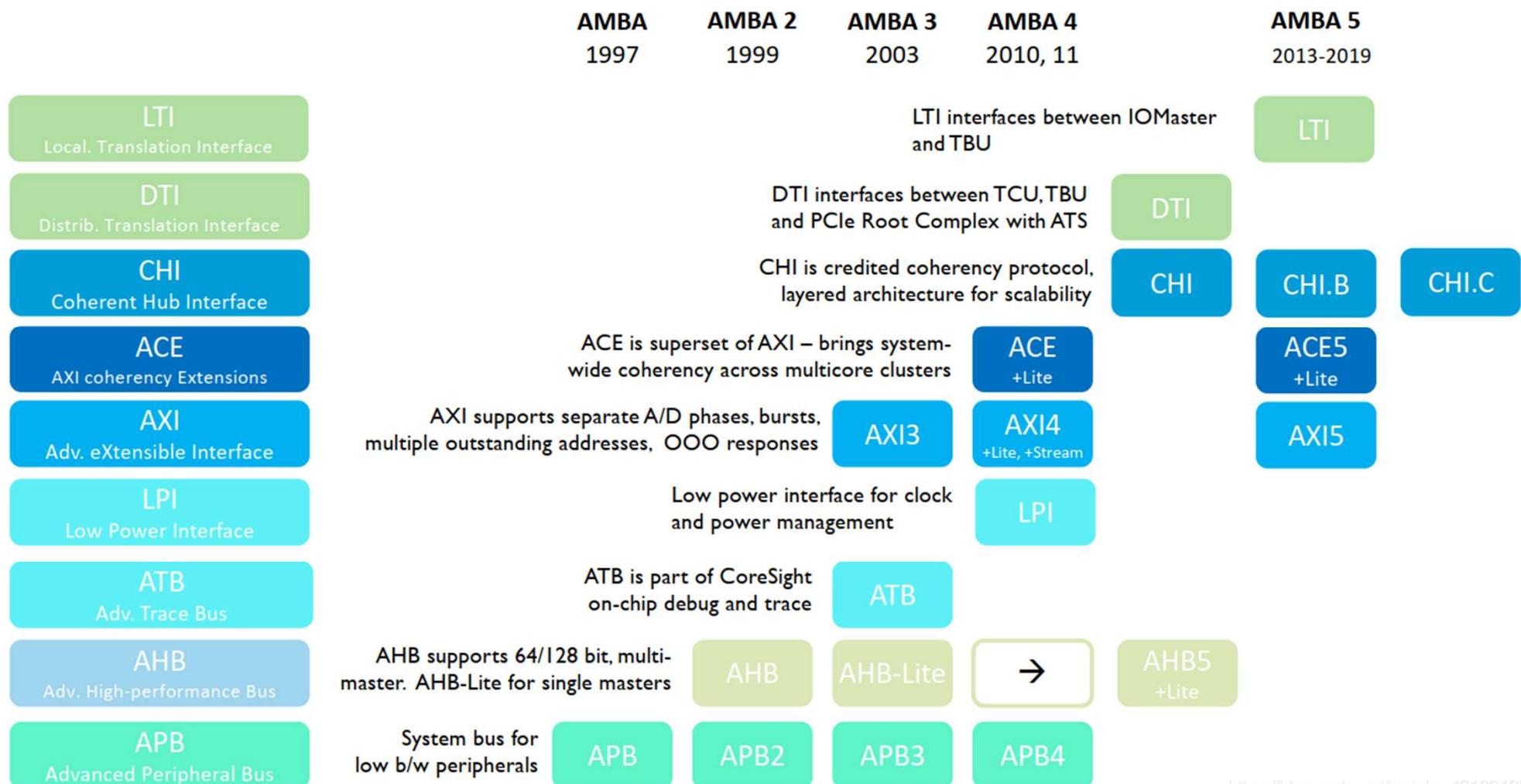
AMBA 3

AMBA 4

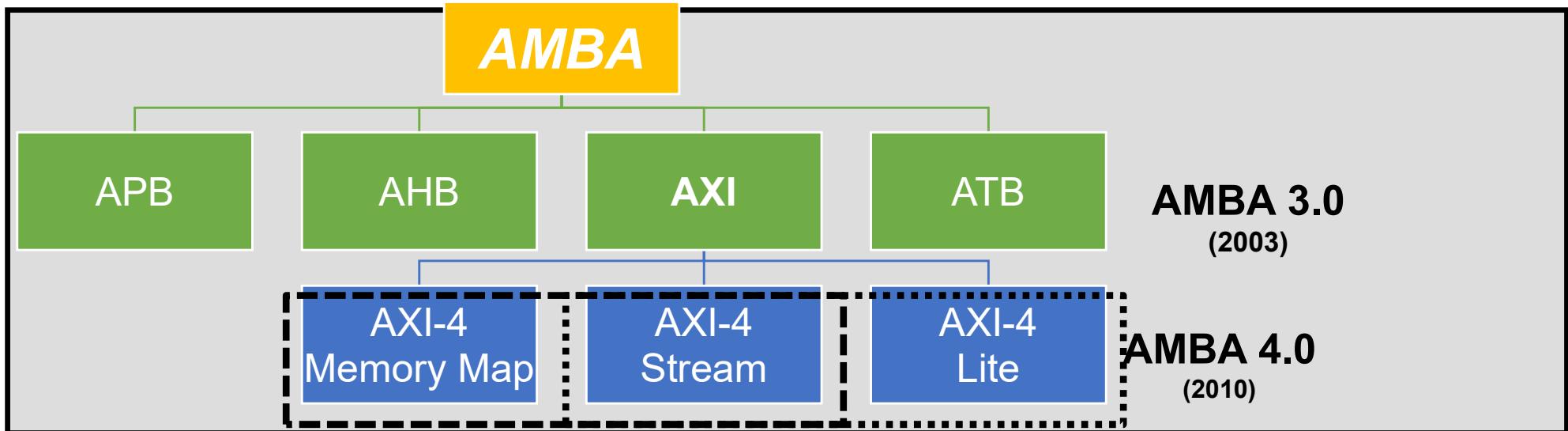
AMBA 5

CHI Coherent Hub Interface	CHI is a credited coherency protocol, layered architecture for scalability		CHI
ACE AXI coherency Extensions	ACE is superset of AXI – brings system-wide coherency across multicore clusters	ACE +Lite	ACE5 +Lite
AXI Advanced eXtensible Interface	AXI supports separate A/D phases, bursts, multiple outstanding addresses, OoO responses	AXI3	AXI4 +Lite, +Stream
AHB Adv. High-performance Bus	AHB supports 64/128 bit, multi-master. AHB-Lite for single masters	AHB	AHB5 +Lite
APB Advanced Peripheral Bus	System bus for low b/w peripherals	APB2	APB3
			APB4

Evolution of the ARM® AMBA® Specifications



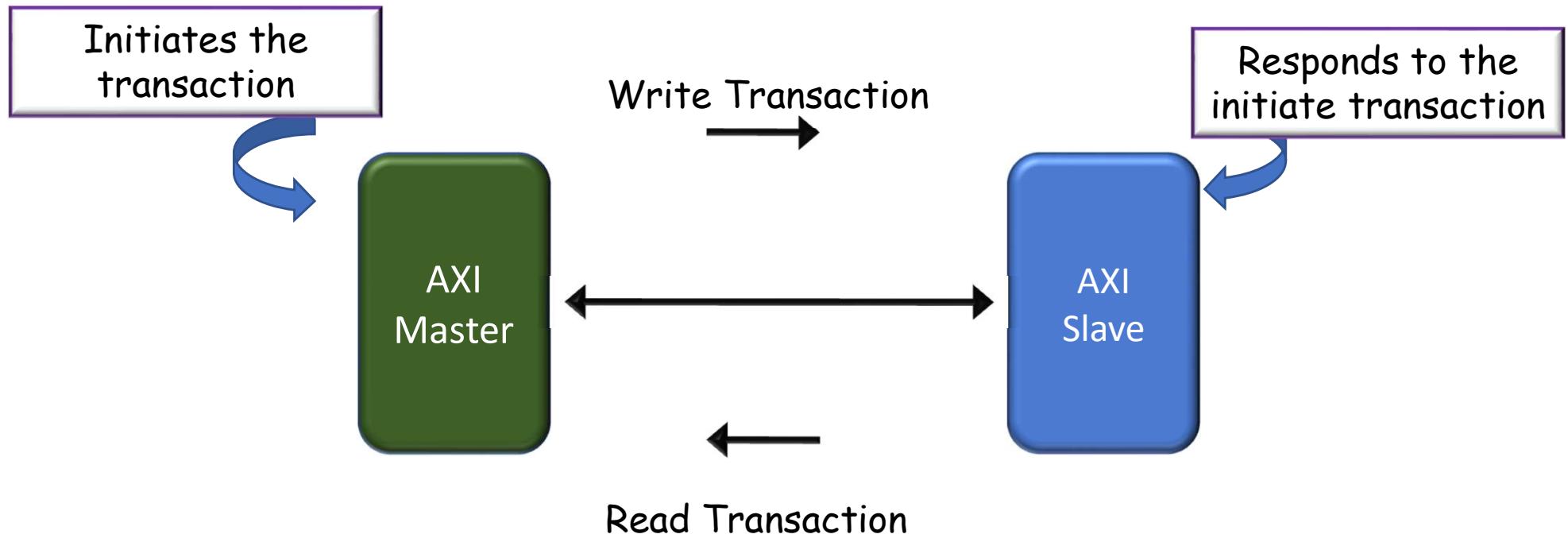
AMBA



Interface	Features	Burst	Data Width	Applications
AXI4	Traditional Address/Data Burst (single address, multiple data)	Up to 256	32 to 1024 bits	Embedded, Memory
AXI4-Stream	Data-Only, Burst	Unlimited	Any Number	DSP, Video, Communications
AXI4-Lite	Traditional Address/Data—No Burst (single address, single data)	1	32 or 64 bits	Small Control Logic, FSM

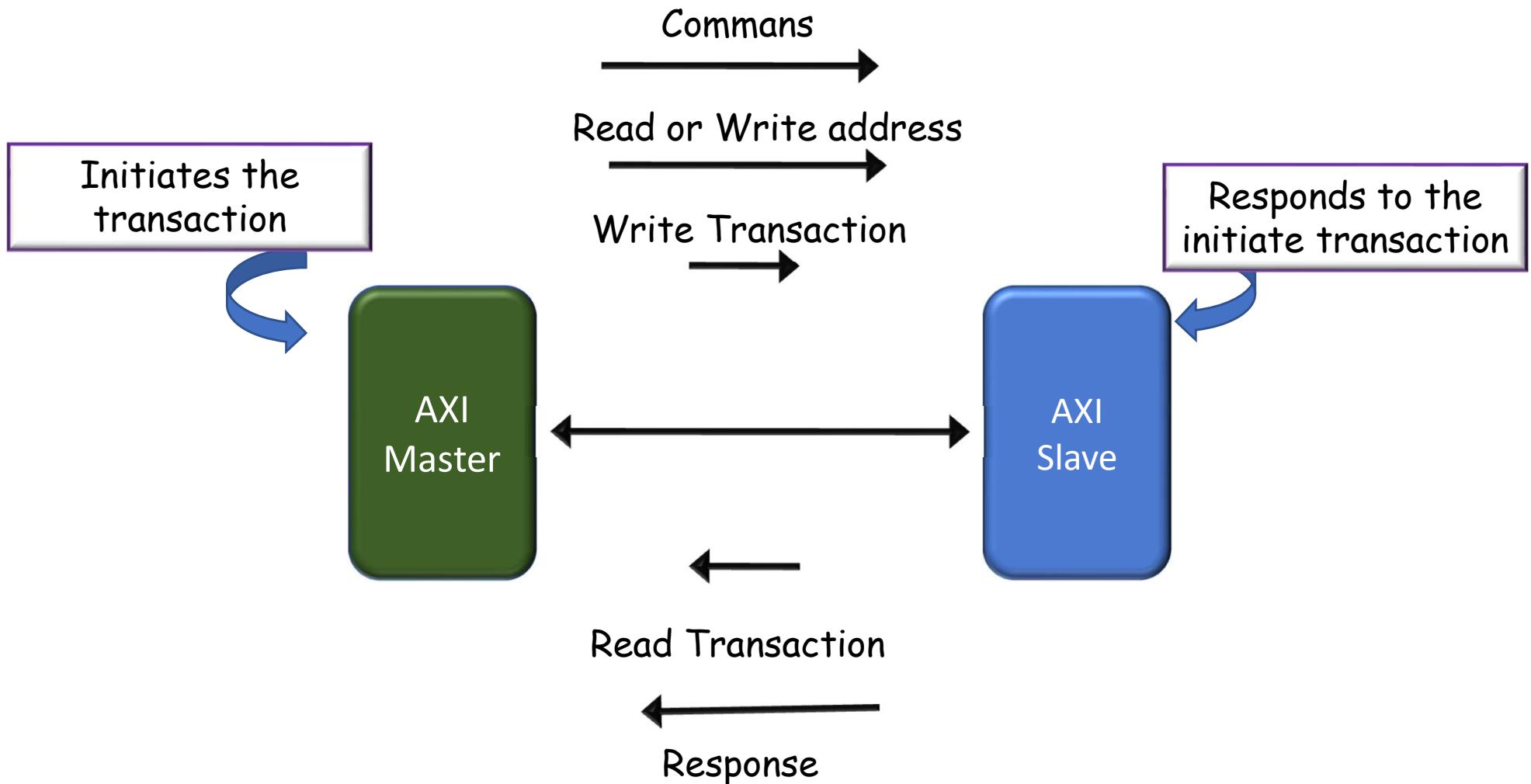
- Vocabulary
 - Channel
 - Independent collection of AXI signals associated to a VALID signal
 - Interface
 - Collection of one or more channels that expose an IP core's connecting a master to a slave
 - Each IP core may have multiple interfaces
 - Transfer
 - Single clock cycle where information is communicated, qualified by a VALID handshake
 - Transaction
 - Complete communication operation across a channel, composed of a one or more transfers
 - Burst
 - Transaction that consists of more than one transfer

AXI Transactions / Master-Slave



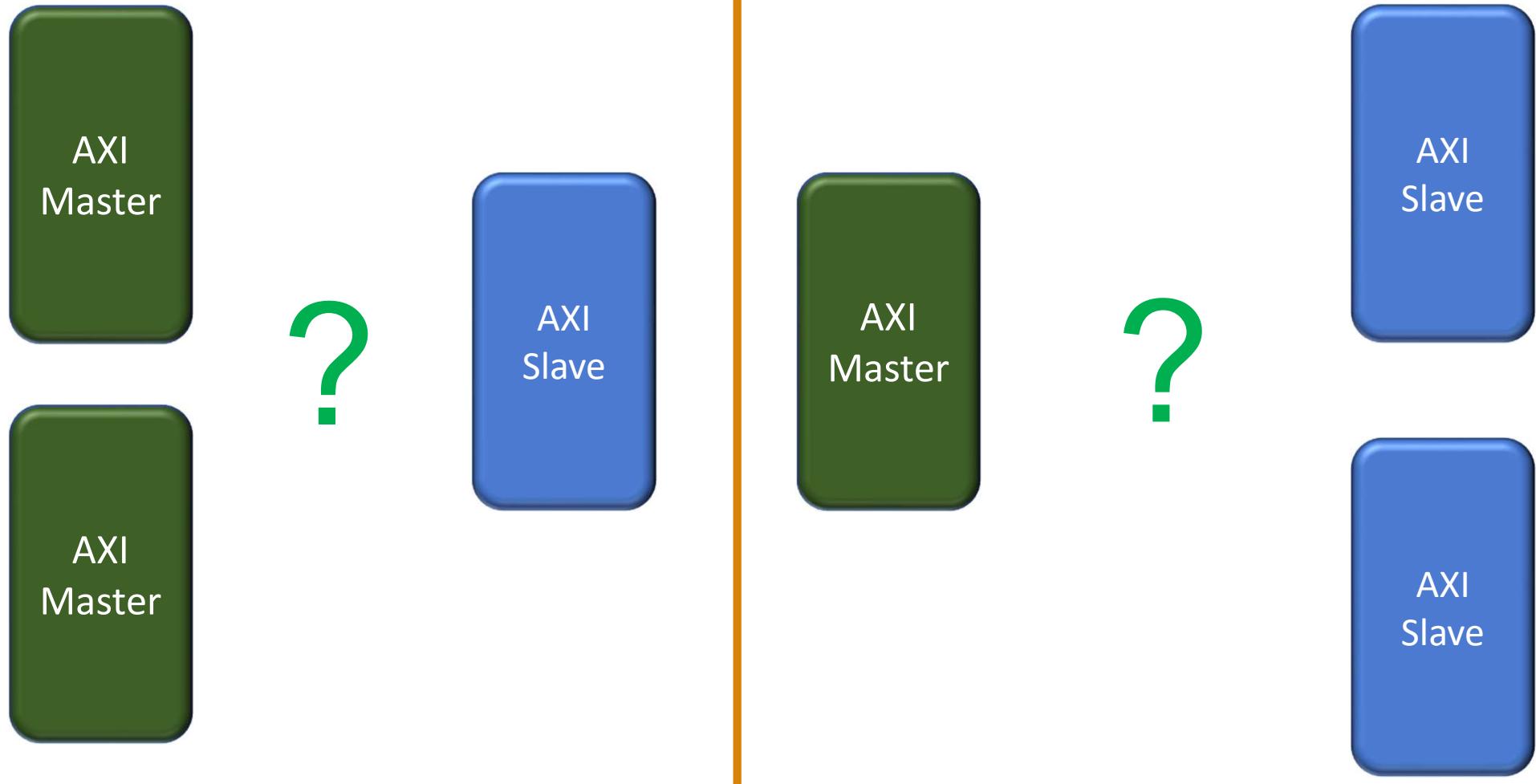
Transactions: transfer of data from one point to another point

AXI Transactions / Master-Slave



Transactions: transfer of data from one point to another point

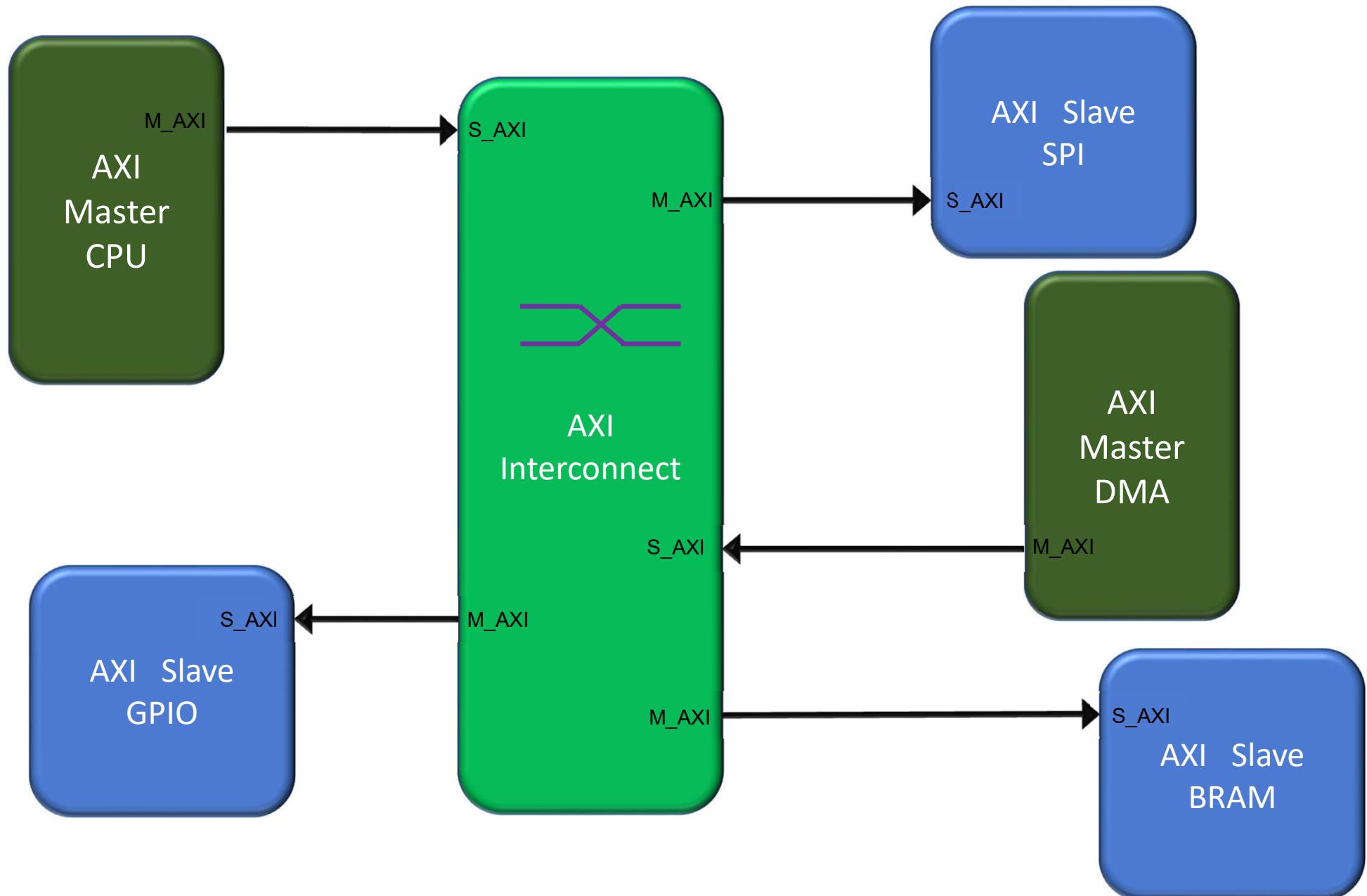
Komunikacija više na više



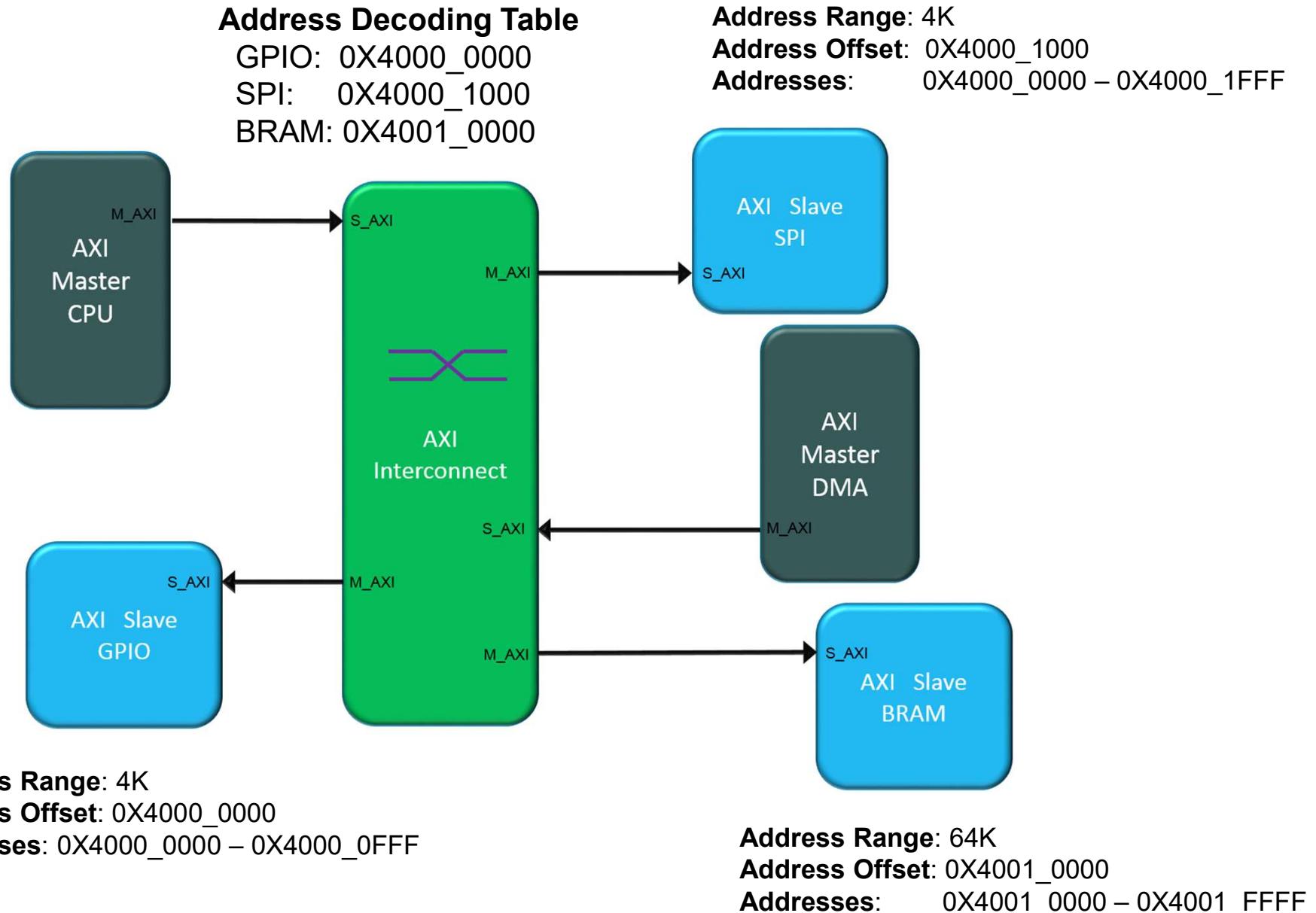
AXI Interconnect

- AXI is an interconnect system used to tie processors to peripherals
 - AXI Full memory map: Full performance bursting interconnect
 - AXI Lite: Lower performance non bursting interconnect (saves programmable logic resources)
 - AXI Streaming: Non-addressed packet based or raw interface

AXI Interconnect



AXI Interconnect – Addressing & Decoding

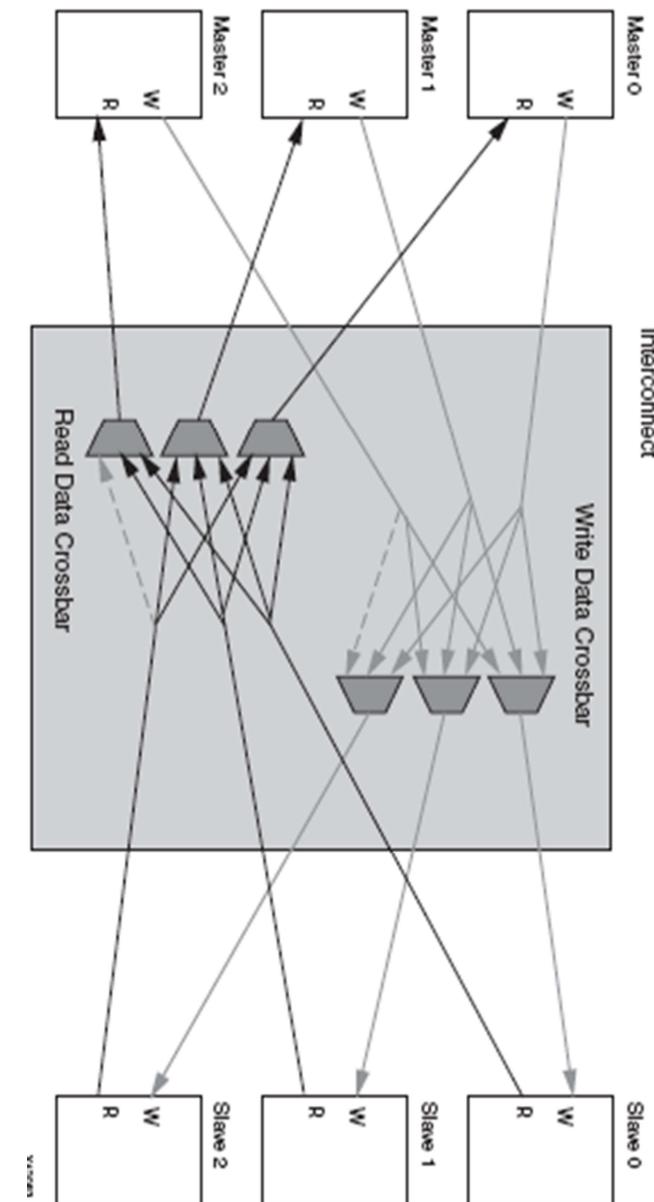


AXI Interconnect

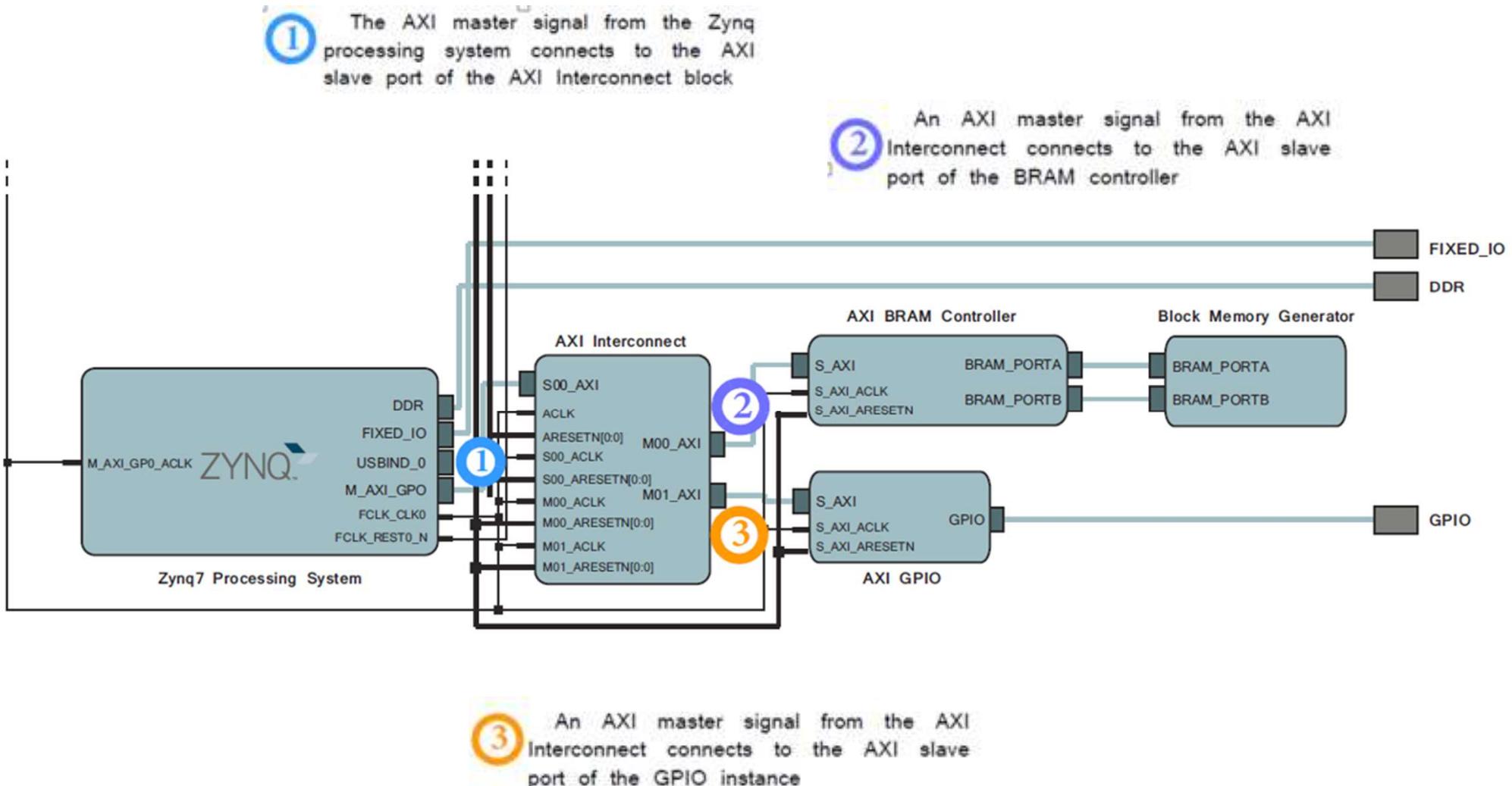
- Different Number of (up to 16)
 - Slave Ports
 - Master Ports
- Data Width Conversion
- Conversion from AXI3 to AXI4
- Register Slices, Input/Output FIFOs
- Clock Domains Transfer

AXI Interconnect

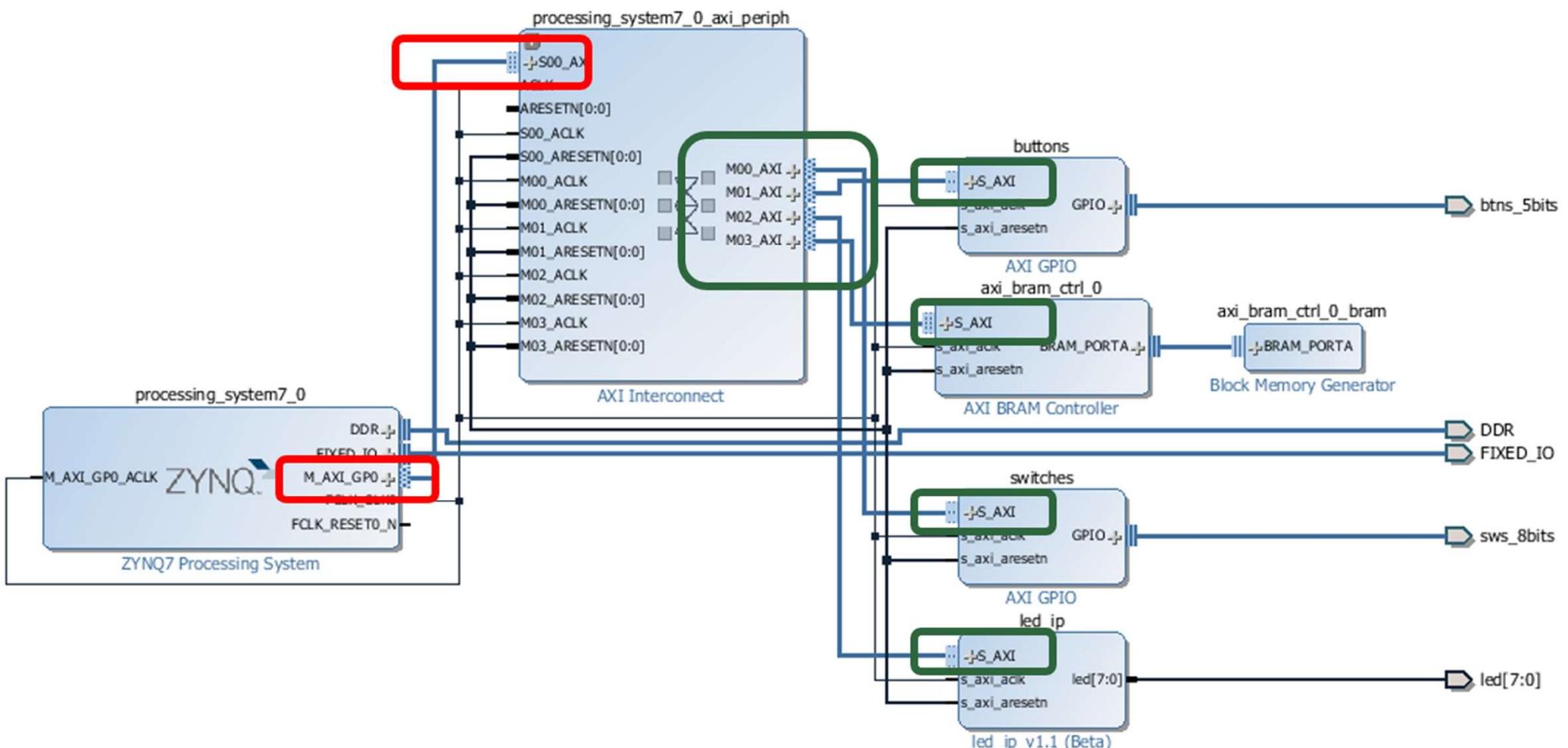
- axi_interconnect component
 - Highly configurable
 - Pass Through
 - Conversion Only
 - N-to-1 Interconnect
 - 1-to-N Interconnect
 - N-to-M Interconnect – full crossbar, shared bus structure
- Decoupled master and slave interfaces
- Xilinx provides three configurable
 - AXI4 Lite Slave
 - AXI4 Lite Master
 - AXI4 Slave Burst
- Xilinx AXI Reference Guide(UG761)



AXI Interface Example



AXI Interface Example



AXI Channels Use A Basic “VALID/READY” Handshake

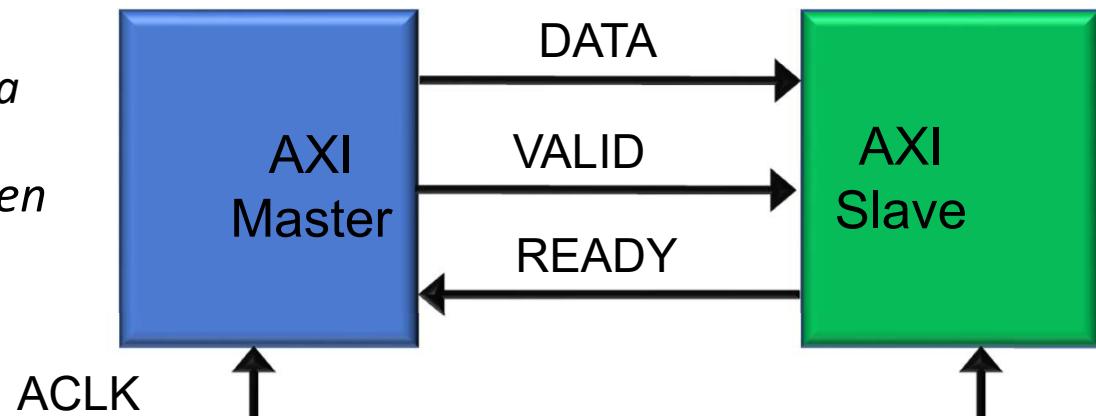
1 Master asserts and hold VALID when data is available

2 Slave asserts READY if able to accept data

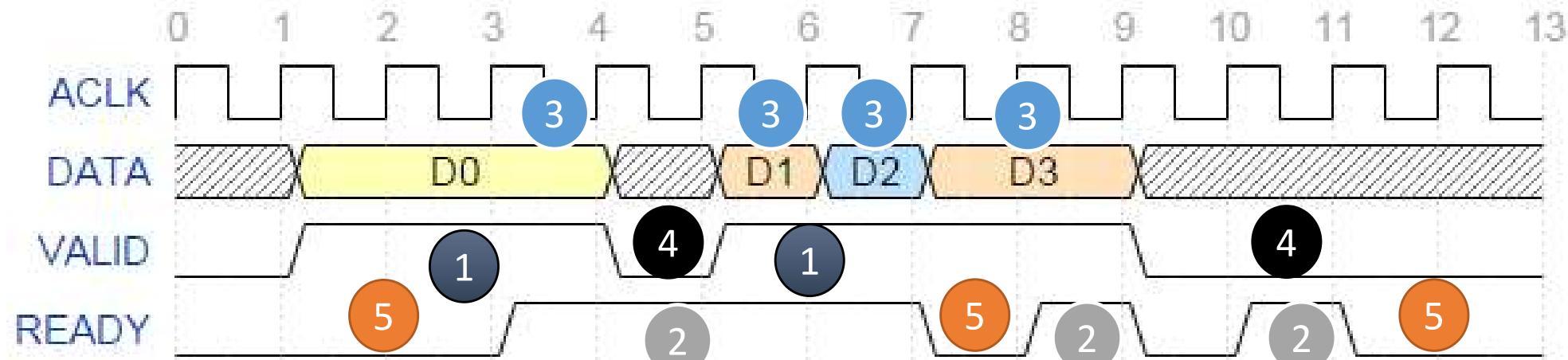
3 Data and other signals transferred when VALID and READY = '1'

4 Master sends next DATA/other signals or deasserts VALID

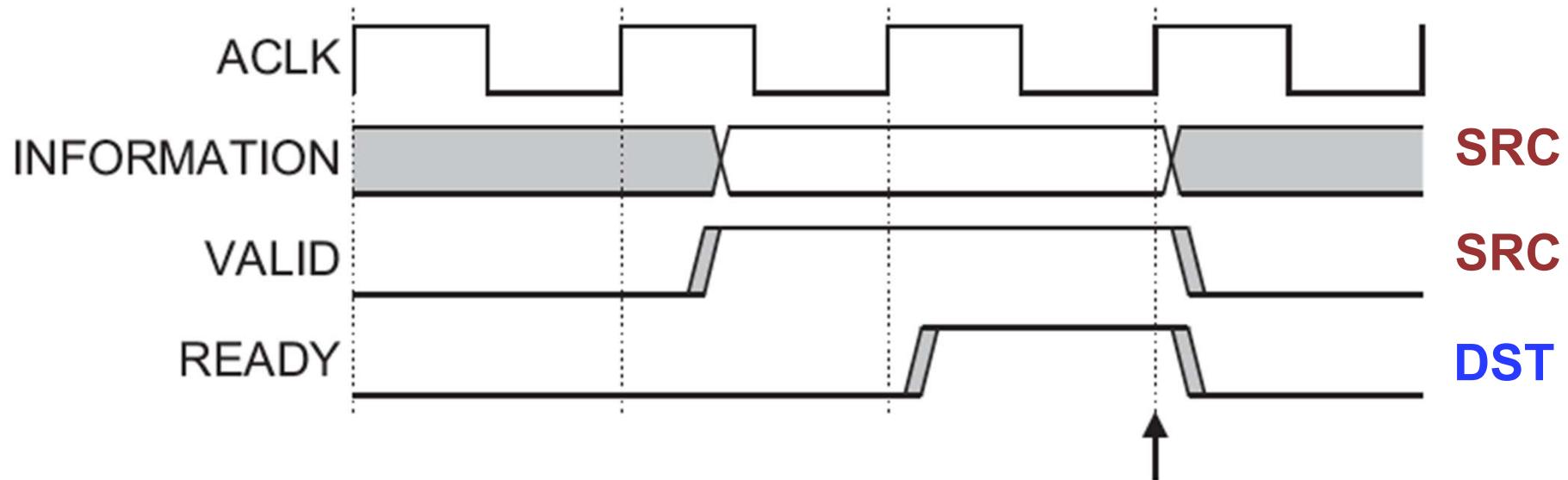
5 Slave deasserts READY if no longer able to accept data



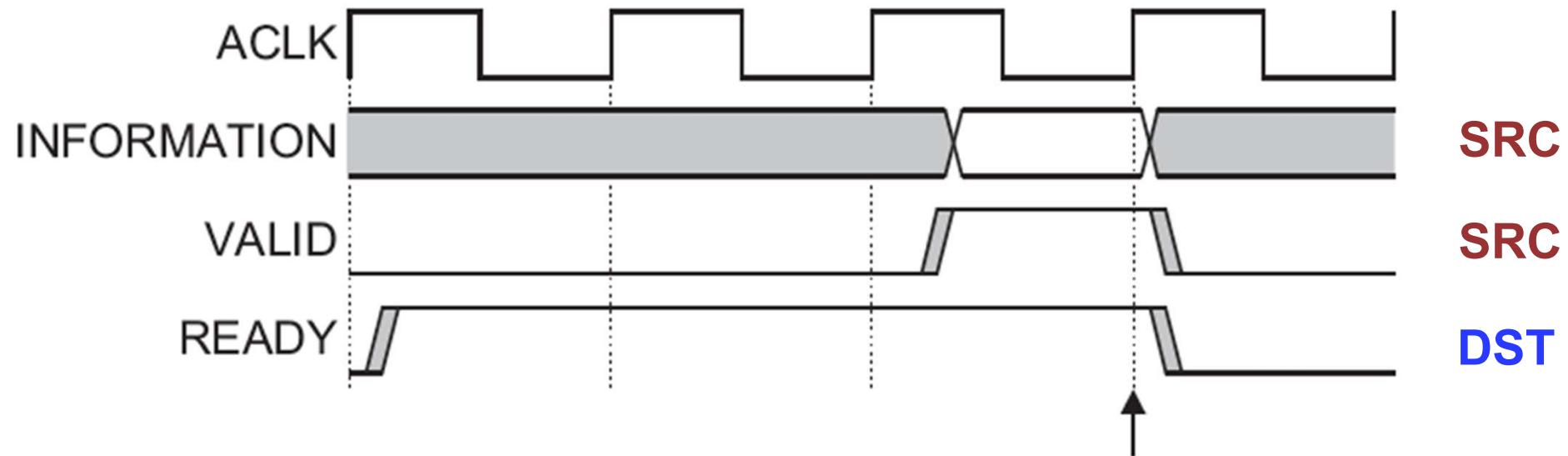
AXI Basic Handshake



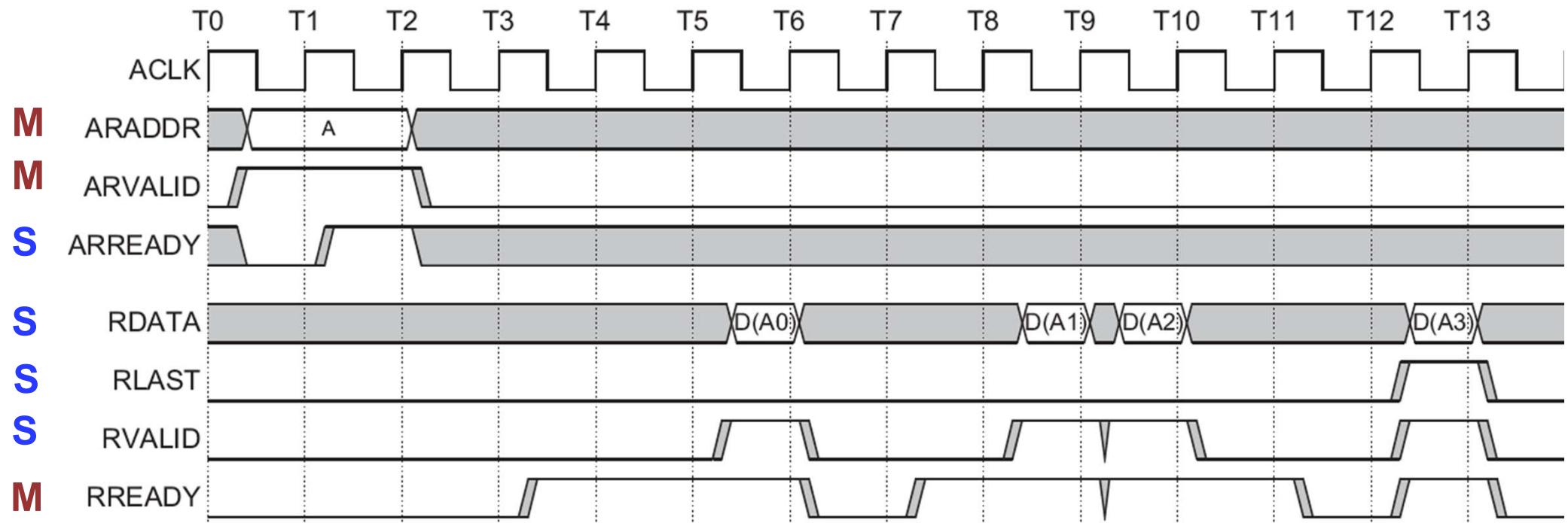
VALID before READY Handshake



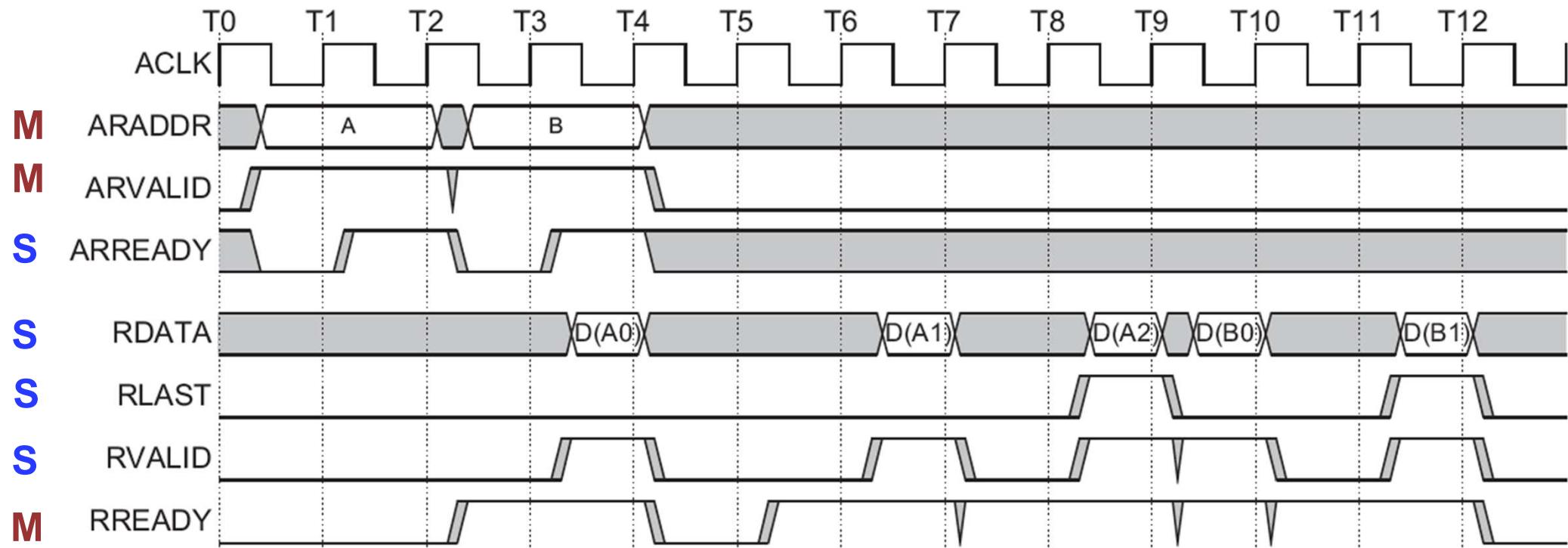
READY before VALID Handshake



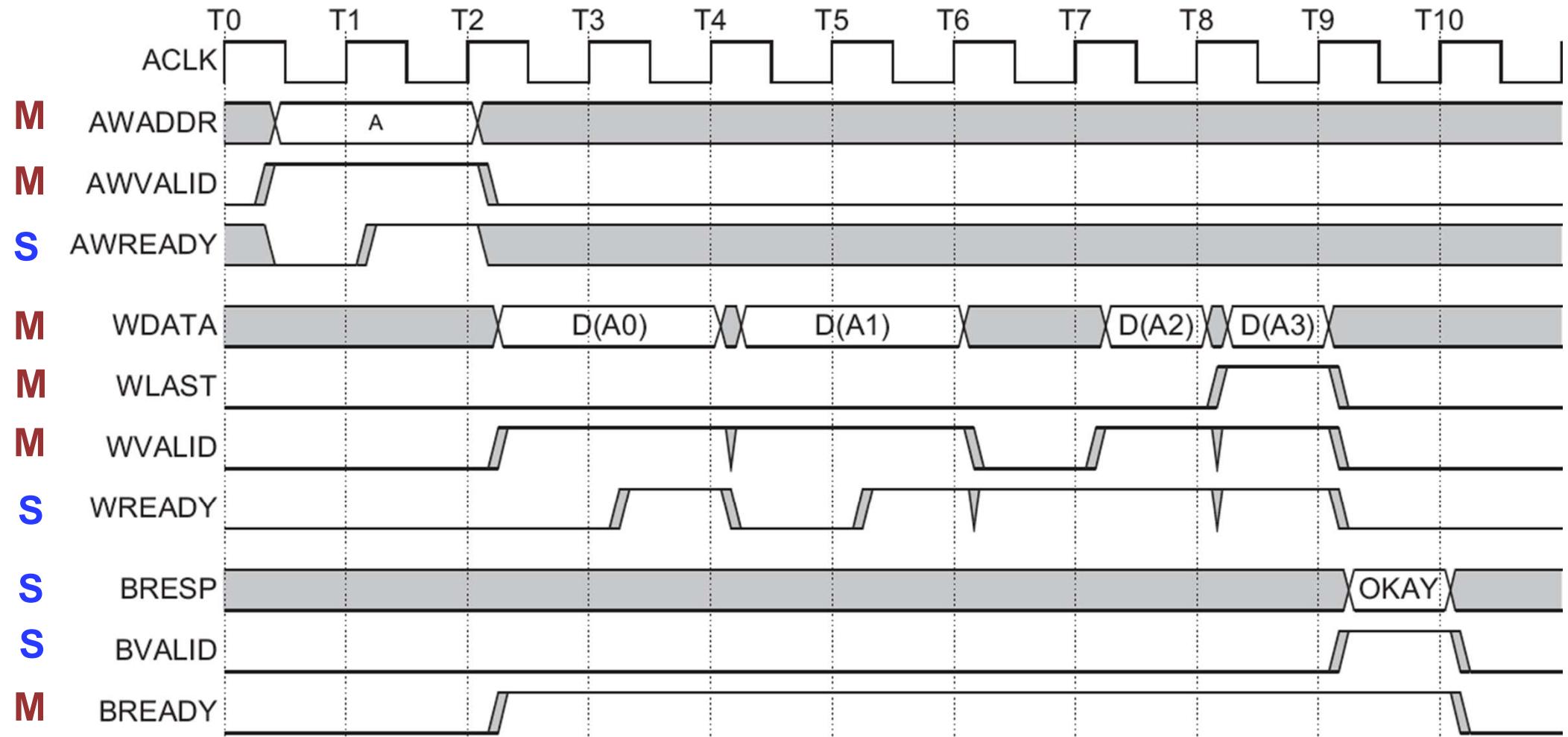
Read Burst



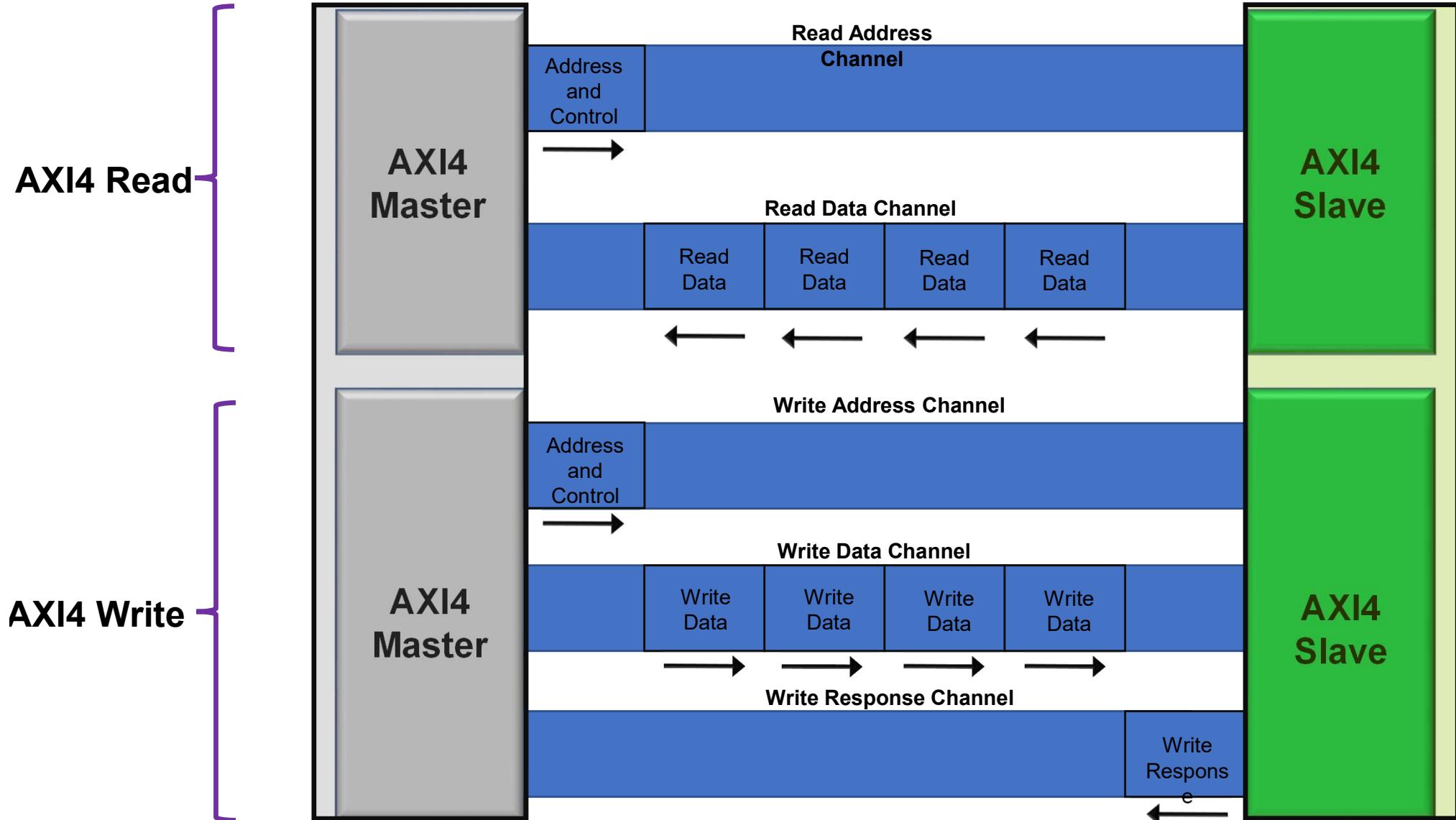
Overlapping Read Bursts



Write Burst

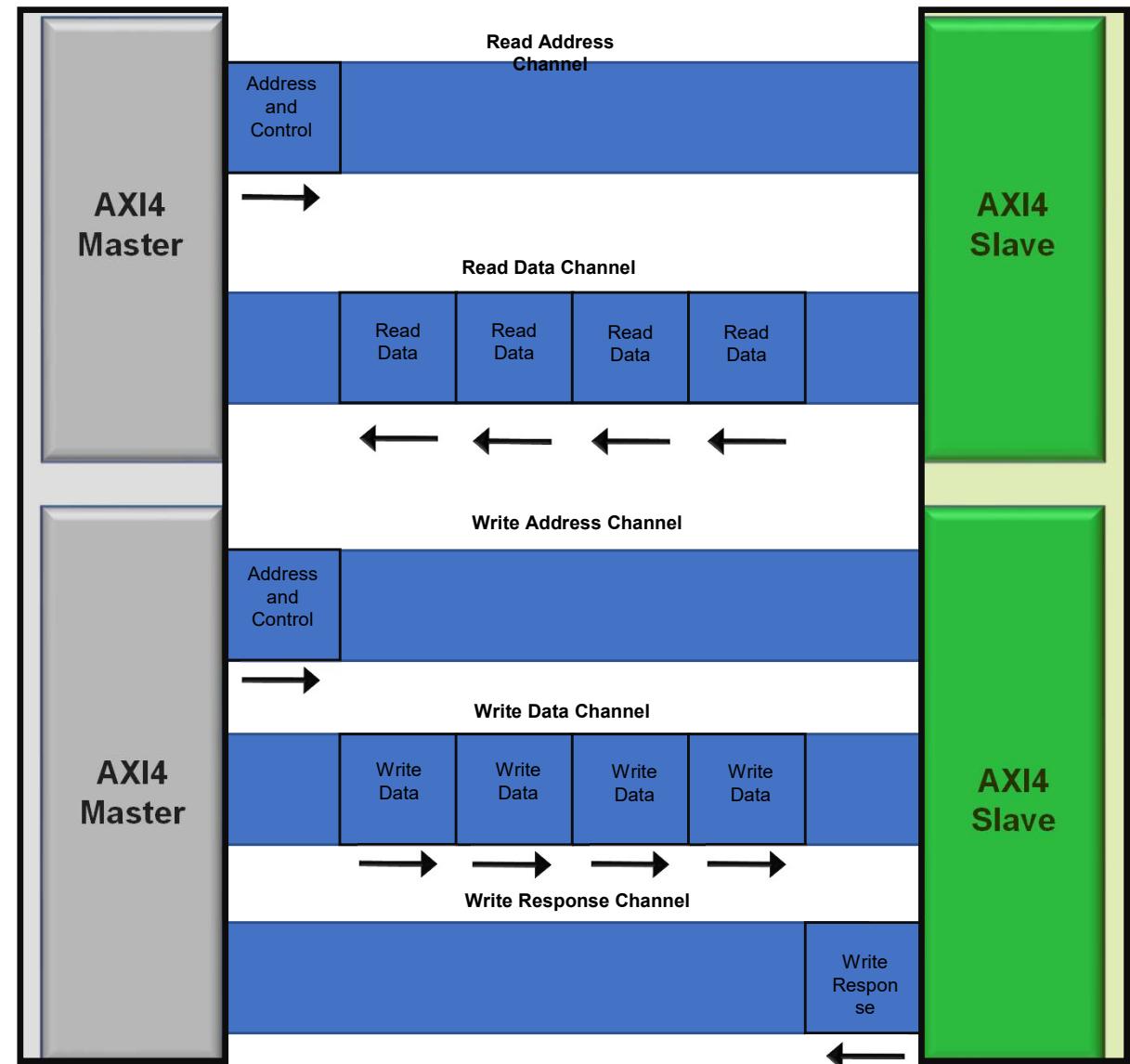


AXI Channels (AXI4 and AXI Lite)



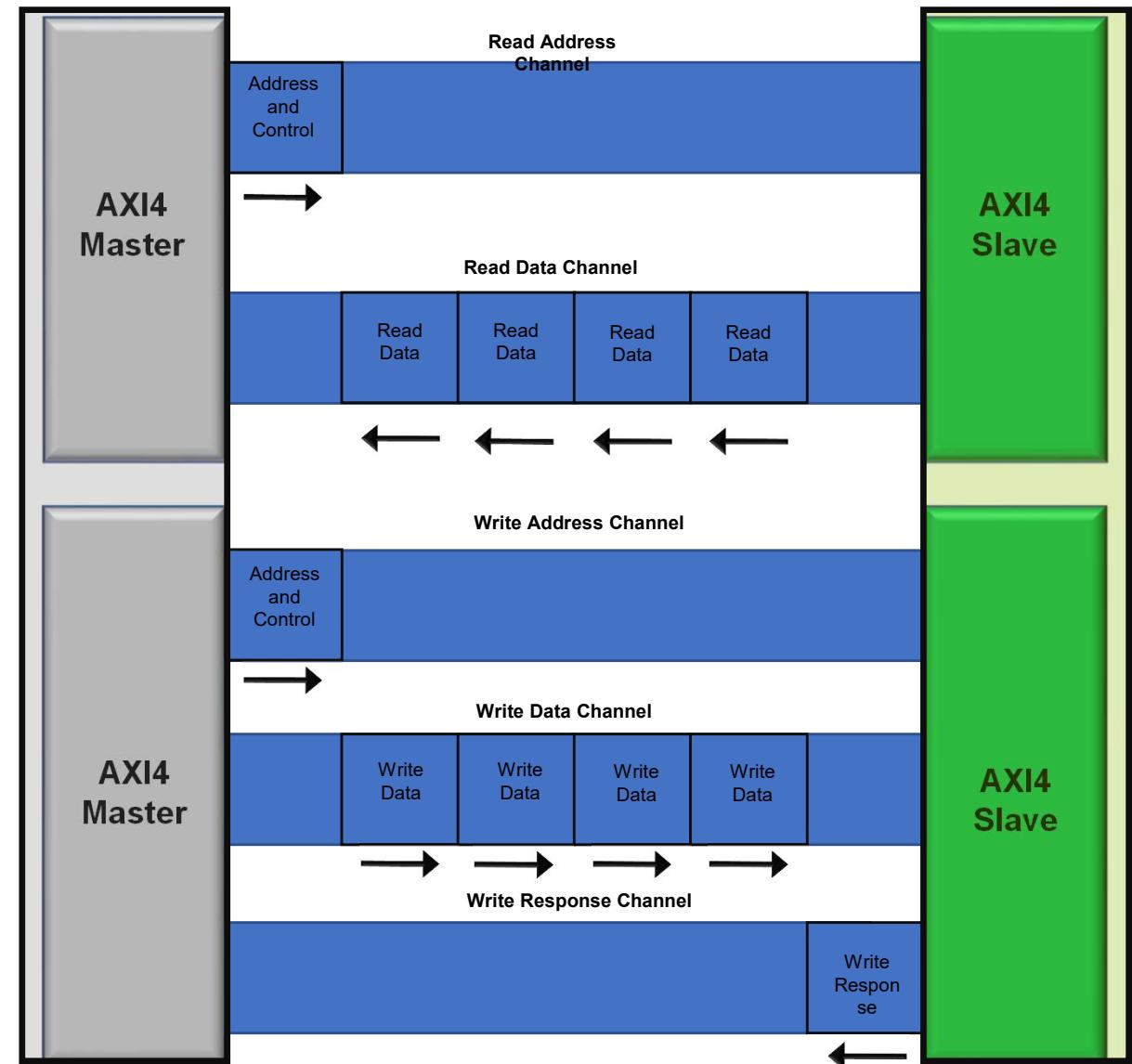
AXI4 - Full

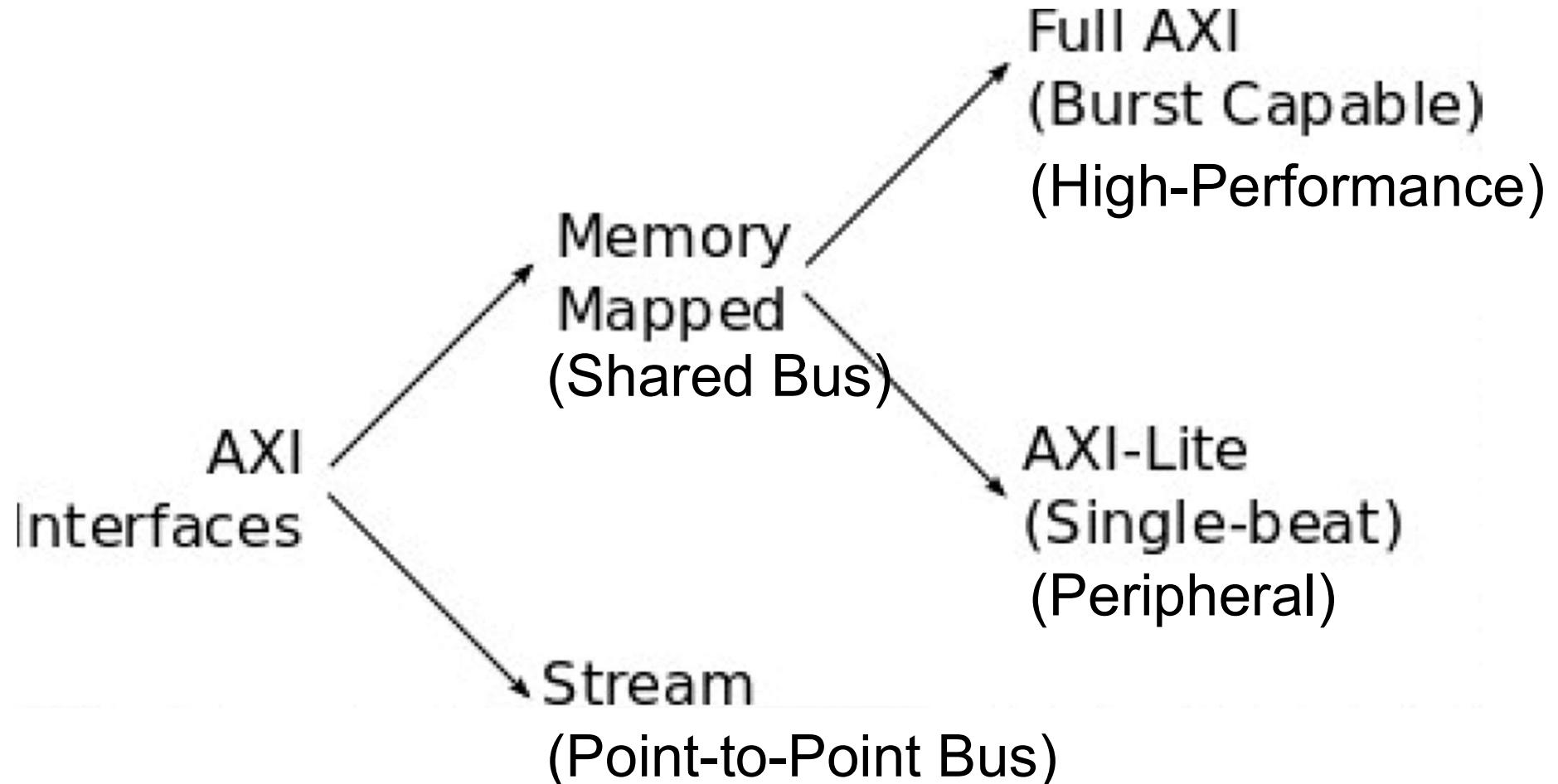
- Sometimes called “*Full AXI*” or “*AXI Memory Mapped*”
- Single address multiple data
 - Burst up to 256 data
- Data Width parameterizable
 - 32, 64, 128, 256, 512, 1024 bits



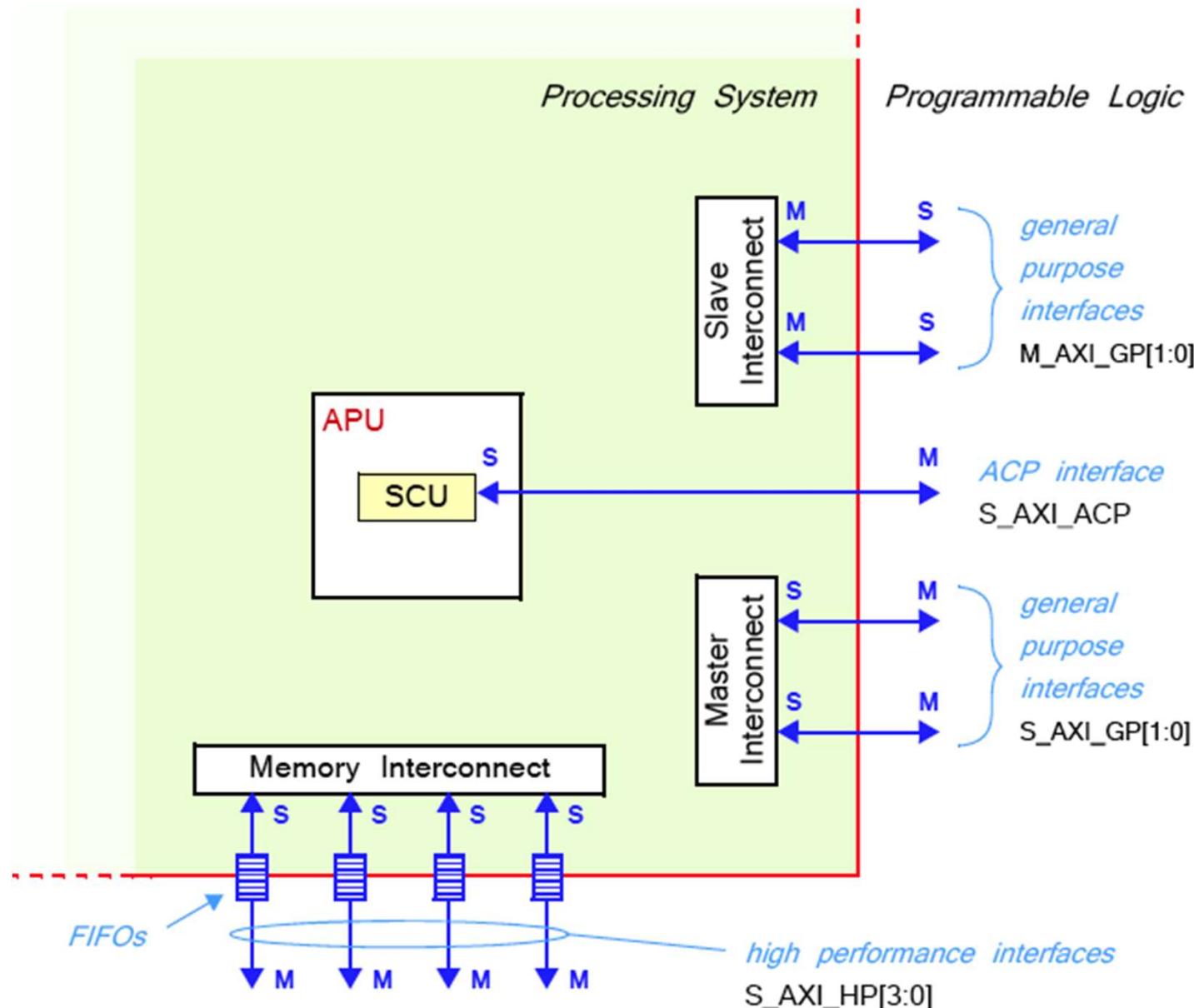
AXI Lite

- No Burst
- Single address, single data
- Data Width 32 or 64 bits (Xilinx IP only support 32)
- Very small size
- The AXI Interconnect is automatically generated





PS-PL Interfaces and Interconnects



AXI Interfaces between PS and PL

Interface Name	Interface Description	Master	Slave
M_AXI_GP0	General Purpose (AXI_GP)	PS	PL
M_AXI_GP1	General Purpose (AXI_GP)	PS	PL
S_AXI_GP0	General Purpose (AXI_GP)	PL	PS
S_AXI_GP1	General Purpose (AXI_GP)	PL	PS
S_AXI_ACP	Accelerator Coherency Port (ACP), cache coherent transaction	PL	PS
S_AXI_HP0	High Performance Ports (AXI_HP) with read/write FIFOs.	PL	PS
S_AXI_HP1	(Note that AXI_HP interfaces are sometimes referred to as AXI Fifo Interfaces, or AFIs).	PL	PS
S_AXI_HP2	(Note that AXI_HP interfaces are sometimes referred to as AXI Fifo Interfaces, or AFIs).	PL	PS
S_AXI_HP3	(Note that AXI_HP interfaces are sometimes referred to as AXI Fifo Interfaces, or AFIs).	PL	PS

General-Purpose Port Summary

- GP ports are designed for maximum flexibility
- Allow register access from PS to PL or PL to PS
- Good for Synchronization
- Prefer ACP or HP port for data transport

High-Performance Port Summary

- HP ports are designed for maximum bandwidth access to external memory and/or On-Chip Memory (OCM)
- When combined can saturate external memory and OCM bandwidth
 - HP Ports : $4 * 64 \text{ bits} * 150 \text{ MHz} * 2 = 9.6 \text{ GByte/sec}$
 - external DDR: $1 * 32 \text{ bits} * 1066 \text{ MHz} * 2 = 4.3 \text{ GByte/sec}$
 - OCM : $64 \text{ bits} * 222 \text{ MHz} * 2 = 3.5 \text{ GByte/sec}$
- Optimized for large burst lengths and many outstanding transactions
- Large data buffers to amortize access latency
- Efficient upsizing/downsizing for 32 bit accesses

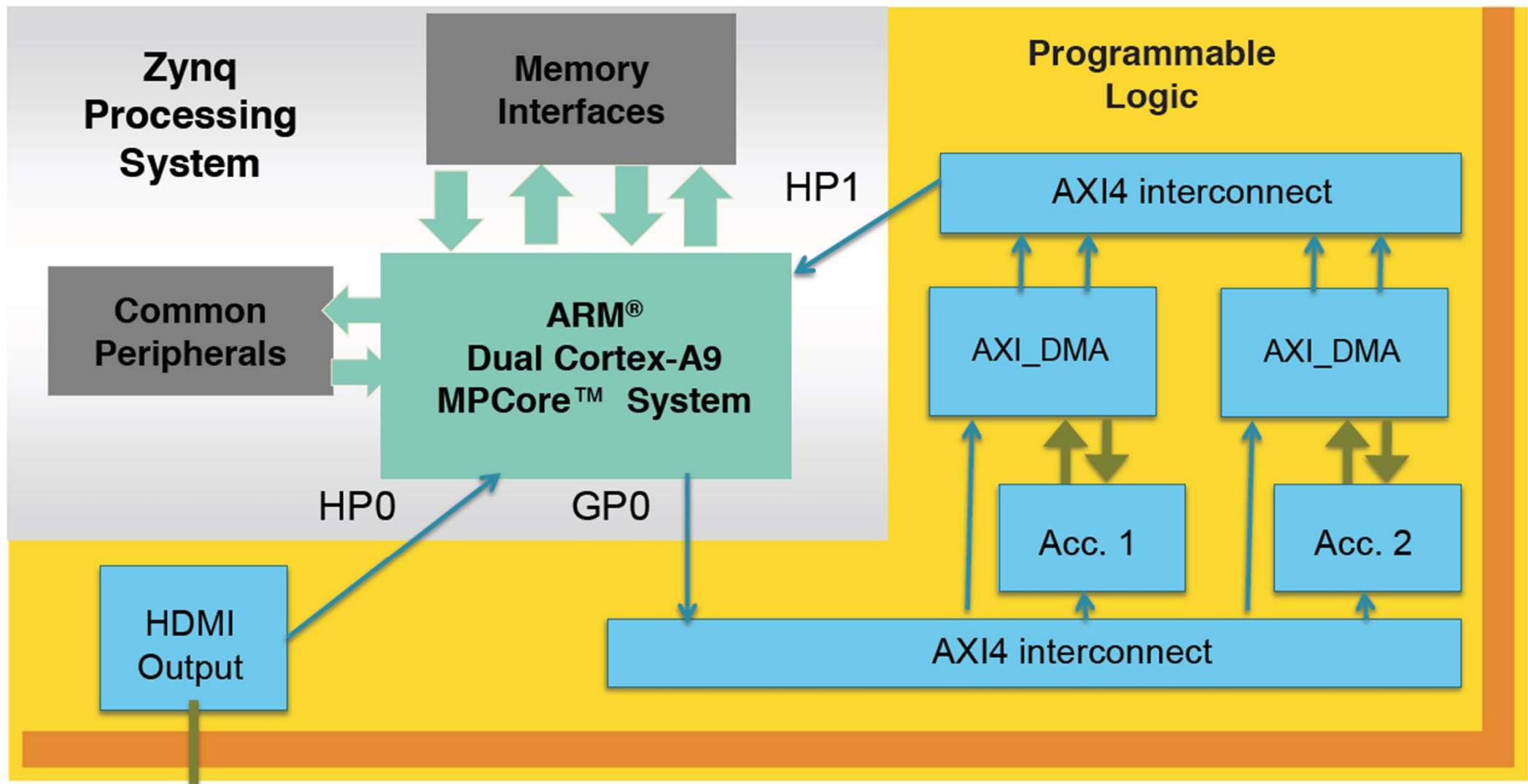
Accelerator Coherency Port (ACP)

- ACP allows limited support for Hardware Coherency
 - Allows a PL accelerator to access cache of the Cortex-A9 processors
 - PL has access through the same path as CPUs including caches, OCM, DDR, and peripherals
 - Access is low latency (assuming data is in processor cache)
no switches in path
- ACP does not allow full coherency
 - PL is not notified of changes in processor caches
 - Use write to PL register for synchronization
- ACP is compromise between bandwidth and latency
 - Optimized for cache line length transfers
 - Low latency for L1/L2 hits
 - Minimal buffering to hide external memory latency
 - One shared 64 bit interface, limit of 8 masters

Accelerator Architecture with DMA

Pro: High Bandwidth Communication

Con: Complicated System Architecture, High Latency



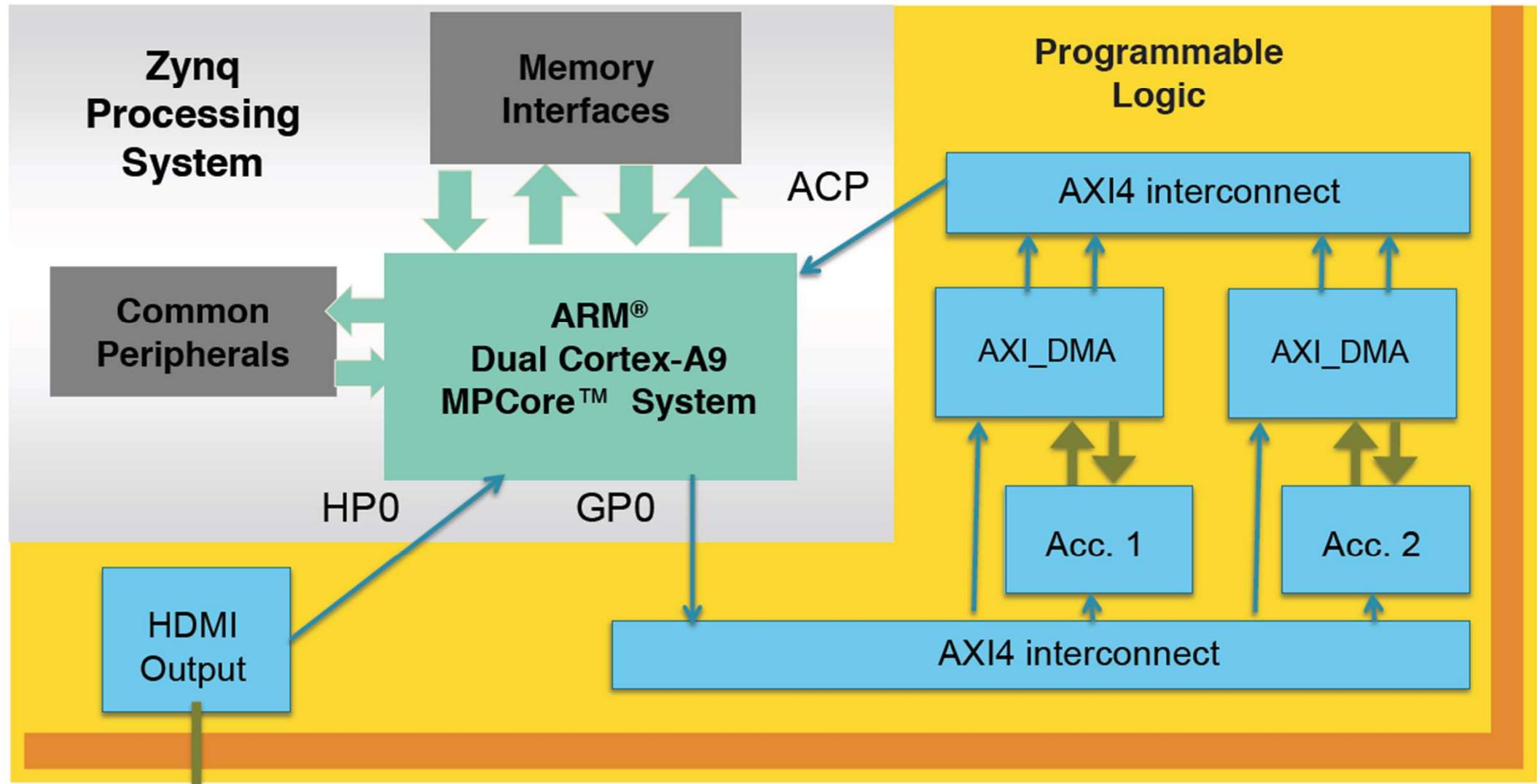
AXI DMA-based Accelerator Communication

- **Write to Accelerator**
 - processor allocates buffer
 - processor writes data into buffer
 - processor flushes cache for buffer
 - processor initiates DMA transfer
- **Read from Accelerator**
 - processor allocates buffer
 - processor initiates DMA transfer
 - processor waits for DMA to complete
 - processor invalidates cache for buffer
 - processor reads data from buffer

Accelerator Architecture with Coherent DMA

Pro: Low latency, high-bandwidth communication

Con: Complicated system architecture, Limited to data that fits in caches

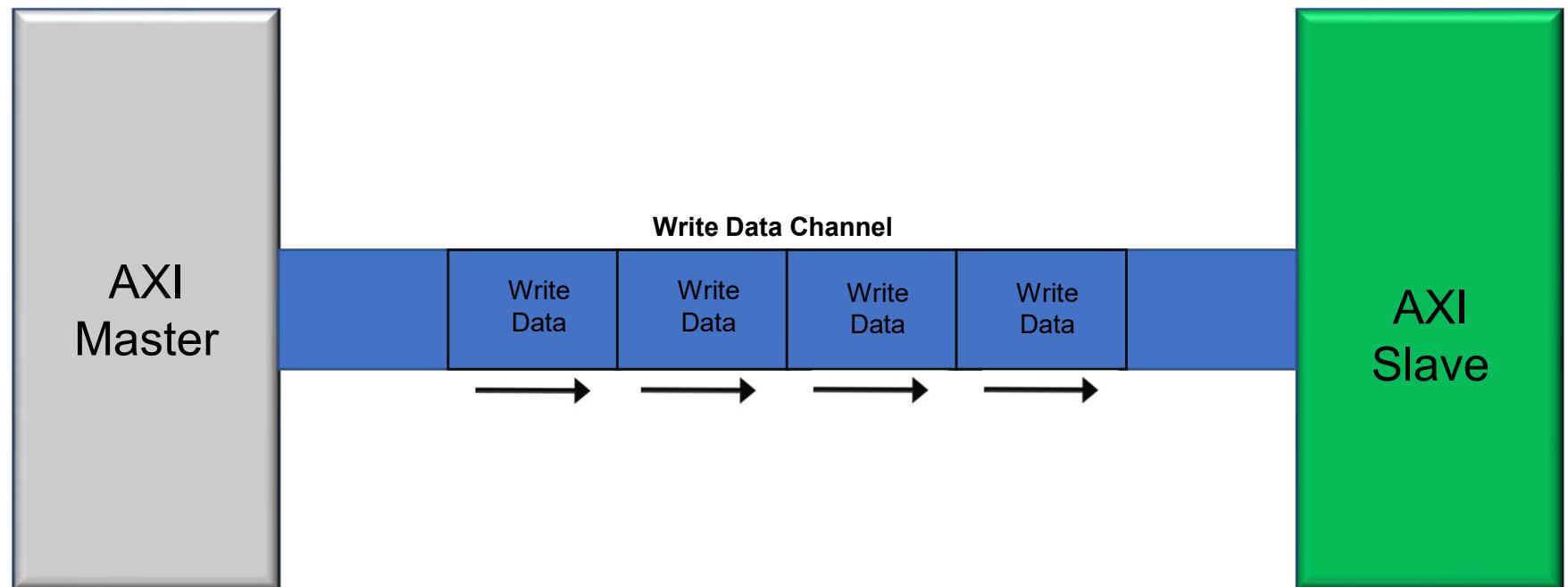


Coherent AXI DMA-based Accelerator Com.

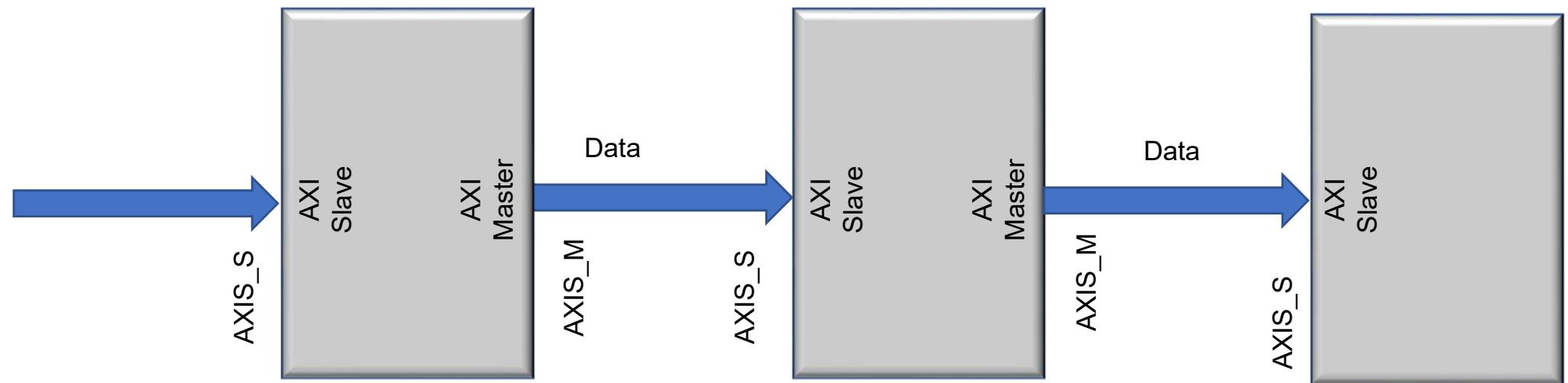
- **Write to Accelerator**
 - processor allocates buffer
 - processor writes data into buffer
 - ~~processor flushes cache for buffer~~
 - processor initiates DMA transfer
- **Read from Accelerator**
 - processor allocates buffer
 - processor initiates DMA transfer
 - processor waits for DMA to complete
 - ~~processor invalidates cache for buffer~~
 - processor reads data from buffer

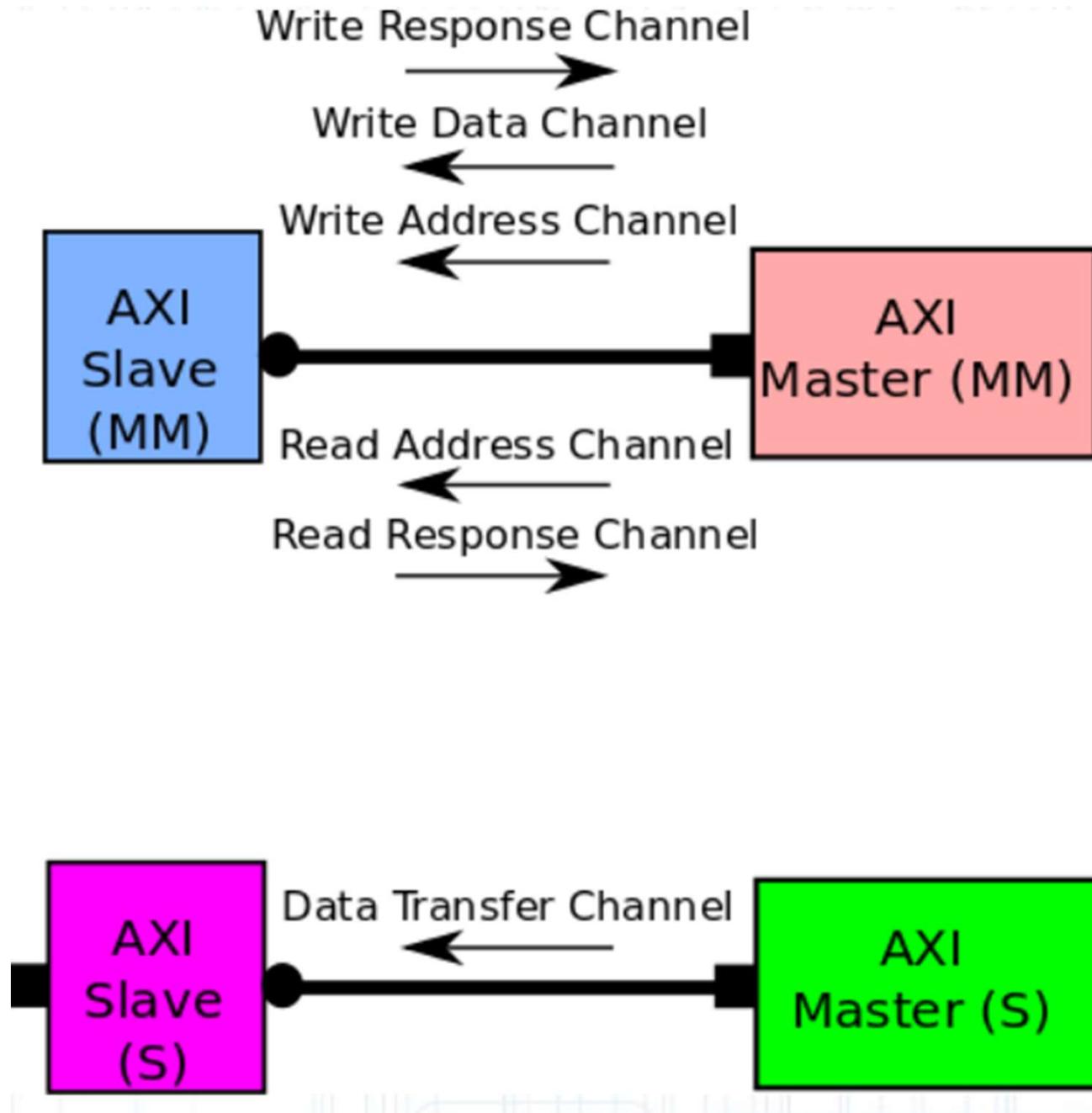
AXI4 Stream

- No address channel, no read and write, always just Master to Slave
 - Just an AXI4 Write Channel
- Unlimited burst length
- Supports sparse, continuous, aligned, unaligned streams

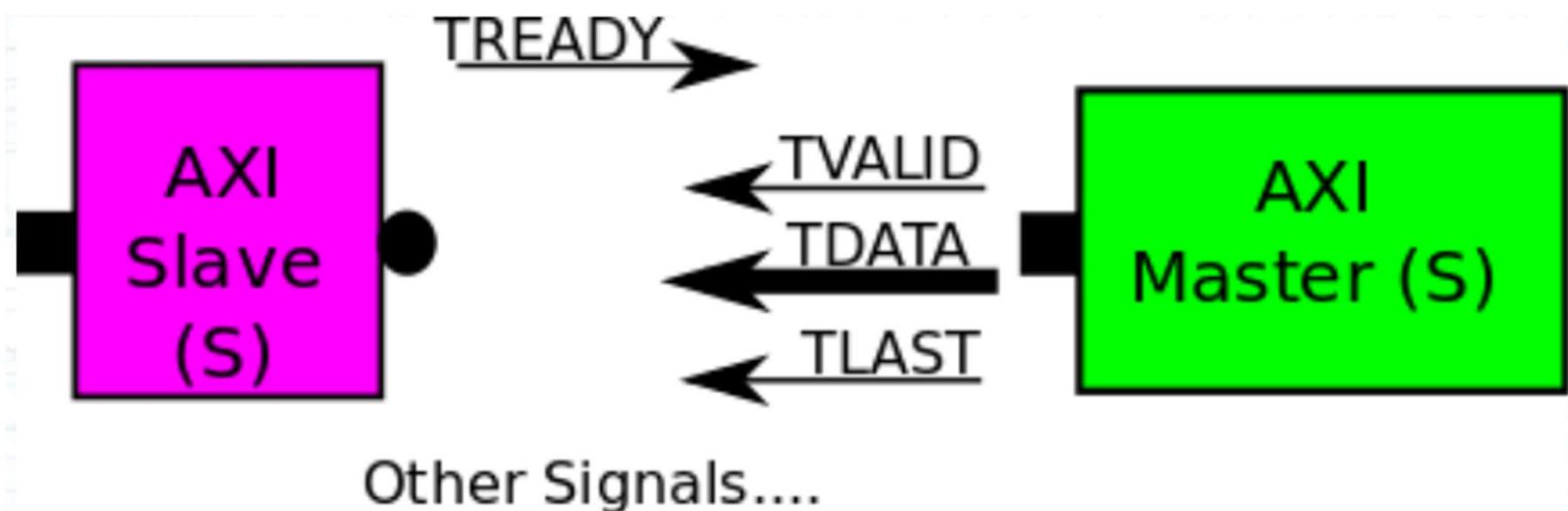


AXI4 Stream



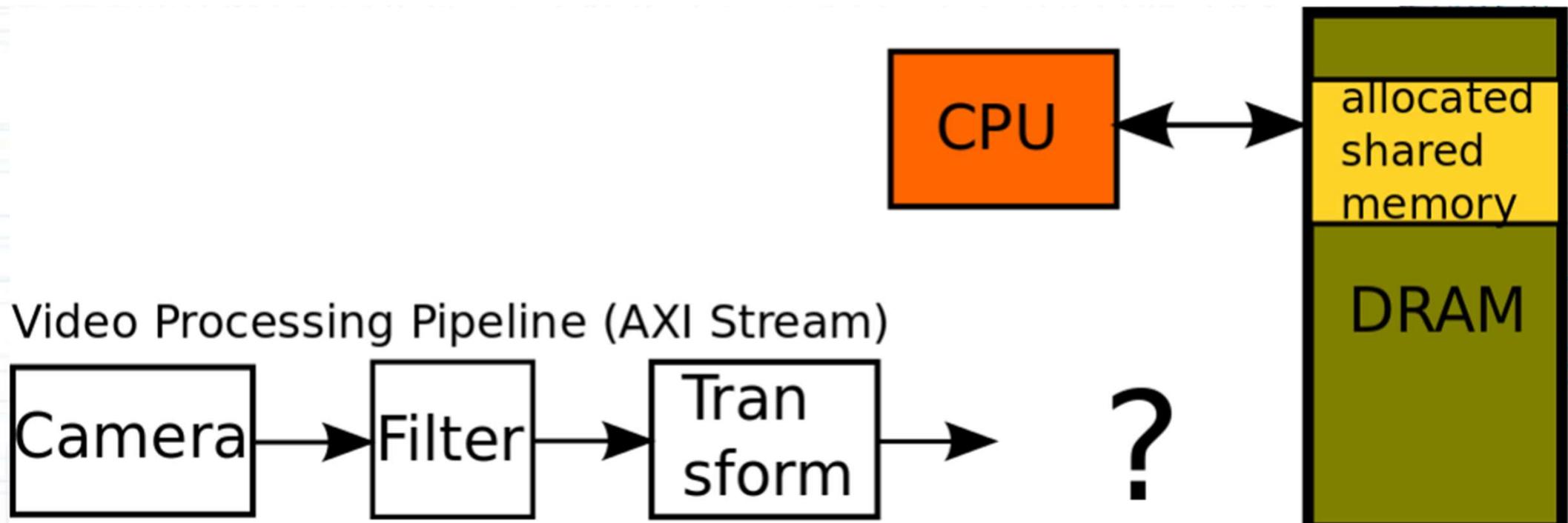


AXI Channel



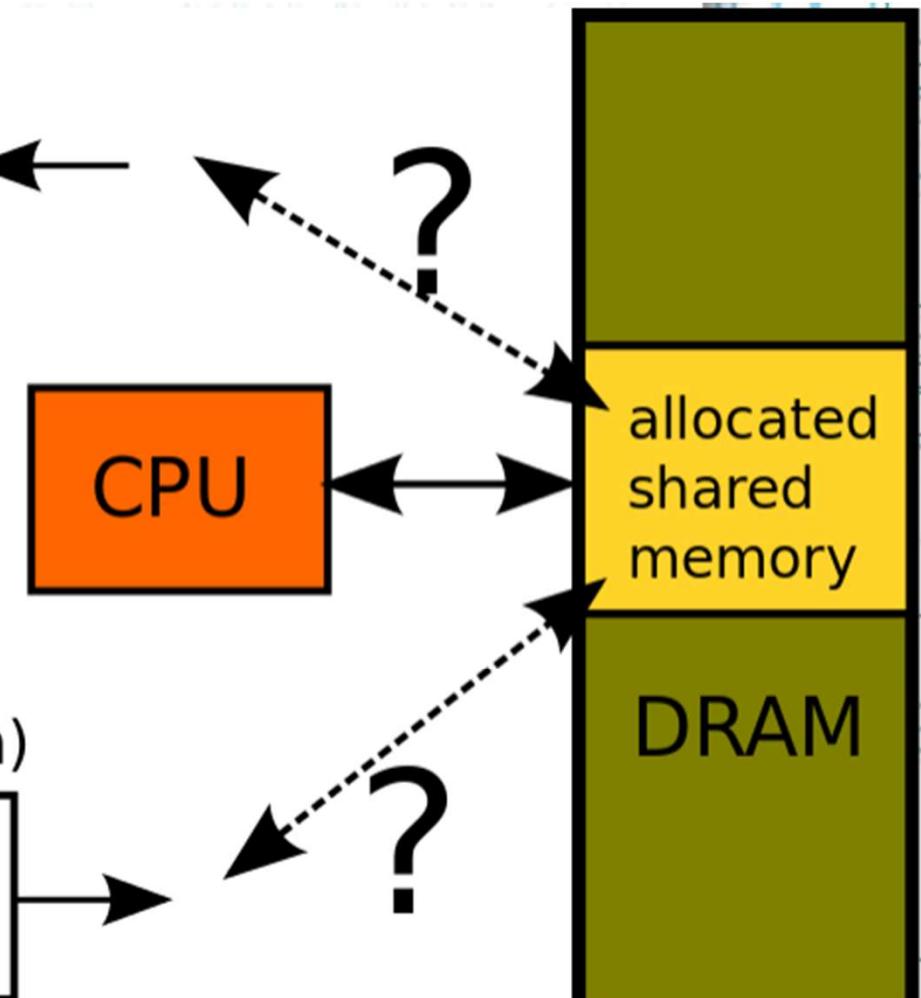
AXI Stream

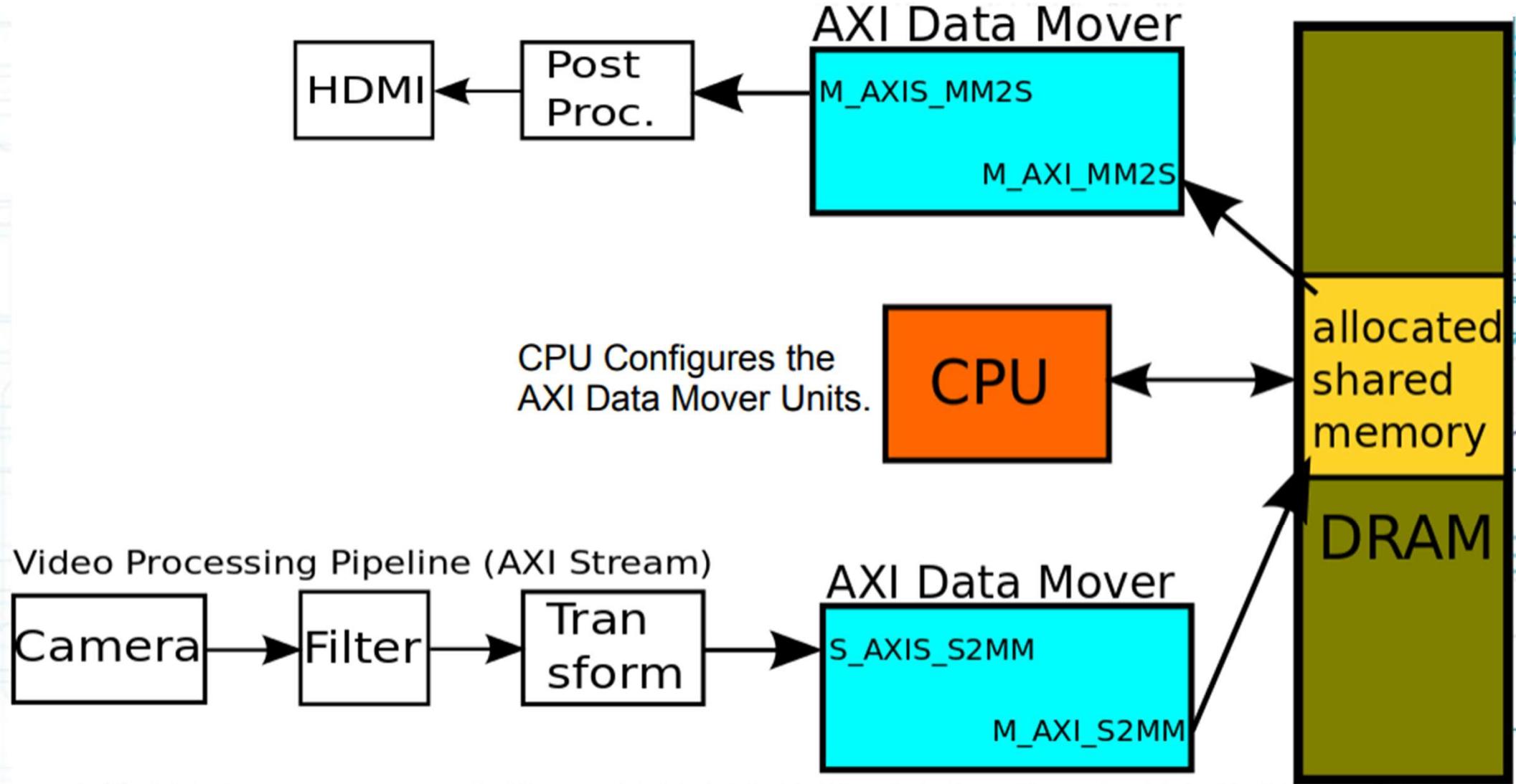
- Output of AXI stream interface
 - Should be finally stored in memory



AXI Lite

Video Processing Pipeline (AXI Stream)





AXI Data Mover

- Gets configured by the CPU
- Interrupts when a transfer task is done (e.g. a frame is transferred completely)
- Gets triggered by the CPU
- Customized Versions:
 - AXI Central DMA engine
 - AXI Video DMA

Naš sustav

