# PCIe, MIPI CSI-2, HDMI and DDR memory across platforms from an Open Source perspective



Alexander Wirthmüller aw@mpsitechnologies.com

# Introduction

### About me

- Based in Munich
- Diploma in Electrical Engineering
- Founder and Director at MPSI Technologies
- MPSI Technologies: make Embedded Software development more fun by replacing repetitive tasks by model-based source code generation
- (Freelance) roles as R&D engineer for cutting-edge projects requiring FPGA-based subsystems one at a time





# The new normal of low-cost FPGA interfaces

- Former justification to use FPGA's: multiple SPI/UART/GPIO now covered by modern MCU's
- Major FPGA use case still is high-bandwidth custom data processing
- Requires high-bandwidth data sources and sinks as well as storage (memory)

#### Why ...

- ... PCI Express?
- L, talk to (PC) host
- ... MIPI CSI-2?
- Any video processing can use a camera input
- ... HDMI (out)?
- quick visualization of video processing results even with (Linux) host present
- ... DDR memory?
- internal BlockRAM too small for megapixel images in mid-range devices

#### Why not ...

- ... Ethernet \*MII?
- in most cases useful only with protocols on top (e.g. UDP/TCP). This is better done from within an OS like Linux
- ... HDMI (in)?
- beyond scope, postponed
- ... USB 3.0?
- h, non-typical FPGA interface, postponed
  ... JESD204\*?
- beyond scope, RF ADC/DAC almost always comes along with very expensive hardware



# Mission statement

### For this presentation and for the underlying effort

- All mid-range devices since 2010 support at least some of the "new normal low-cost" FPGA interfaces
- BUT: low-cost doesn't imply low-effort (making "low-cost" relative)
- AND: these interfaces are one major obstacle for switching between vendors easily



- Collect and publish information on interfaces at a public location on the web, keep it up-to-date
- Offer Open Source wrappers which make interface access vendor-neutral
- Identify Open Source software and vendor-provided examples which can be re-used for other vendors' platforms
- Implement a fully documented and easy-to-reproduce reference application on all suitable platforms
- Do not look at advanced features rather: block data transfer with PCIe, standard image streaming with MIPI, HD monitor output with HDMI, block read/write from/to DDR memory
- Focus on affordable mid-range devices, limit custom hardware to adapter PCB's



# Mid-range FPGA landscape \*)

Vendor	Product line	Year (node)	LE (range)	CPU	PCle	MIPI	HDMI 1)	DDR
	Artix 7	2011 (28nm)	13k-215k	-	yes	ext. pass.	yes	PHY muxed
AMD (Xilinx)	Zynq	2011 (28nm)	23k-444k	ARMv7	some	ext. pass.	yes	PHY dedic. 2)
	Zynq UltraScale+ CG	2015 (16nm)	81k-600k	ARMv8	some	I/O std.	yes	PHY dedic. 2)
Efinix	Titanium	2021 (16nm)	36k-176k 3)	soft RISC-V	no 3)	hard IP	yes	PHY dedic.
	Cyclone V SE	2012 (28nm)	25k-110k	ARMv7	no	ext. pass.	yes	PHY dedic. 2)
intelPSG (Altera)	Cyclone V GX	2011 (28nm)	36k-301k	-	yes	ext. pass.	yes	PHY muxed
	Agilex 5	2023 ("7"nm)	50k-656k	ARMv8	Yes	TBD	yes	TBD
Lattice	CrossLink-NX	2020 (28nm)	17k-39k	-	some	hard IP	yes	I/O feat.
Microchip (-semi)	PolarFire SoC	2019 (28nm)	23k-461k	RISC-V	yes	I/O std.	yes	PHY dedic. 2)

1) HDMI only requires 165MHz LVDS for full HD, 2) shared with CPU complex, some with additional PHY muxed,

3) larger capacities and PCIe-capable SerDes announced

\*) non-exhaustive, "cost optimized" in some vendors' slang Information from vendor sources retrieved online in June 2023

PCIe, MIPI CSI-2, HDMI and DDR memory across platforms from an Open Source perspective



# A typical mid-range project Original version

#### Hardware features

- Turntable with stepper motor
- Tripod with camera/laser holder
- OV5640 five megapixel camera with parallel interface
- Two adjustable red line lasers





Calibration



#### Software features

- Red line laser on/off difference imaging for trace detection
- Orientation via Harris corner detection algorithm
- Point cloud calculated on host
- Control via web UI from host

PCIe, MIPI CSI-2, HDMI and DDR memory across platforms from an Open Source perspective



# A typical mid-range project Original AMD Zynq 7010 implementation



#### **Limitations**

- OV5640 sensor with parallel output only on market (and obsolete)
- On-FPGA storage allows for one bit per pixel at full resolution
- Not even possible to copy whole frames to host



low < 500 kB/s

medium < 50MB/s

high > 50MB/s

# A typical mid-range project

Adding extras / Digilent Nexys Video with AMD Artix 7 implementation





# A typical mid-range project

## Adding extras / Digilent Nexys Video with AMD Artix 7 implementation

#### **Improvements**

- Compatible with many MIPI cameras offering multiple resolution choices
- DDR memory allows HDR image capture (multi exposure times to counter laser saturation)
- DDR memory plus PCIe allows transferring whole frames to host
- HDMI output allows for quick verification without host







#### Digilent Arty Z7-20 (arty)

- Based on Zynq Z7020
- \$299 (digilent.com)
- Device XC7Z020-1CLG400C
- €128 (mouser.de)
- Host: Yocto Linux on ARMv7
- X PCIe
- ✓ MIPI CSI-2 via HDMI in
- ✓ HDMI out
- ✓ 512MB DDR3 SDRAM



	🚞 _mdl	>	i arty	· · · · · ·	Arty.vhd
	in _rls	>	igsk 📄 cgsk	>	Arty.xdc
wskd >	.editorconfig		illeb 🚞 cleb	>	Axislave_v1_0.vhd
	🫅 .git	>	iccl	>	Bcdfreq_v1_0.vhd
	.gitignore		imcep	>	Camacq.vhd
	i ezdevwskd	>	inxsv 🚞	>	Camif.vhd
Col.	fpgawskd	>	🚞 tidk	>	Crc8005_32.vhd
	LICENSE		🚞 zudk	>	Debounce_v1_0.vhd
and a second	mcuwskd	>			Featdet.vhd
	README.md				Hostif.vhd
R					l2cmaster_v1_0.vhd
					Ident.vhd
					Laser.vhd
					Parrom.vhd
The second second					Bgbled_v1_0.vhd
the second s					Spimaster_v1_0.vhd
					State.vhd
					Step.vhd
					Timeout_v1_0.vhd
					Tkclksrcy_v1_0.vhd
					Top.vhd
					Uart Easy v1 0.vhd

Digilent Arty Z7-20 (arty)

- Based on Zynq Z7020
- \$299 (digilent.com)
- Device XC7Z020-1CLG400C
- €128 (mouser.de)
- Host: Yocto Linux on ARMv7

X PCIe

- ✓ MIPI CSI-2 via HDMI in
- ✓ HDMI out
- ✓ 512MB DDR3 SDRAM

PCIe, MIPI CSI-2, HDMI and DDR memory across platforms from an Open Source perspective





#### Digilent Nexys Video (nxsv)

- Based on Artix 7
- \$549 (digilent.com)
- Device XC7A200T-1SBG484C
- €277 (mouser.de)
- Host: PC / NUC
- ✓ PCle via FMC
- ✓ MIPI CSI-2 via FMC
- ✓ HDMI out
- ✓ 512MB DDR3 SDRAM

PCIe, MIPI CSI-2, HDMI and DDR memory across platforms from an Open Source perspective





#### Avnet ZUBoard 1CG (zudk)

- Based on Zynq UltraScale+ 1CG
- €149 (de.farnell.com)
- Device ZU1CG-1SBVA484I
- €290 (!) (mouser.de)
- Host: Yocto Linux on ARMv8

X PCIe

- ✓ MIPI CSI-2 via SYZYGY TXR
- X HDMI out
- ✓ 1GB DDR4 SDRAM



# FPGA(-SoC) project variants



#### Efinix Titanium Ti180 dev kit (tidk)

- €776 (digikey.de)
- Device Ti180M484I3
- Price N/F
- Host: Buildroot Linux on soft RISC-V

#### X PCle

- ✓ MIPI CSI-2 via SYZYGY
- ✓ HDMI out via FMC + SYZYGY(-like)
- ✓ 32MB LPDDR4 SDRAM

PCIe, MIPI CSI-2, HDMI and DDR memory across platforms from an Open Source perspective



# FPGA(-SoC) project variants intelPSG (Altera)



#### Aries Cyclone V SE SoM (mcep)

- MCV-6DB SoM on EVP (base board)
- €309 SoM only (aries-embedded.de)
- Device 5CSEBA6U23C7N
- €272 (!) (digikey.de)
- Host: Yocto Linux on ARMv7

X PCle

- ✓ MIPI CSI-2 via HSMC
- X HDMI out
- IGB DDR3 SDRAM

PCIe, MIPI CSI-2, HDMI and DDR memory across platforms from an Open Source perspective



# FPGA(-SoC) project variants intelPSG (Altera)



#### terasIC Cyclone V GX st. kit (cgsk)

- €259 (digikey.de)
- Device 5CGXFC5C6F27C7N
- €227 (!) (digikey.de)
- Host: PC / NUC
- ✓ PCIe via HSMC
- ✓ MIPI CSI-2 via HSMC
- ✓ HDMI out
- ✓ 512MB LPDDR2 SDRAM



# FPGA(-SoC) project variants



# Lattice CrossLink-NX eval. board (cleb)

- €118 (digikey.de)
- Device LIFCL-40-9BG400C
- €64 (digikey.de)
- Host: PC / NUC
- ✓ PCle via FMC
- ✓ MIPI CSI-2
- X HDMI out
- X DDR memory



# FPGA(-SoC) project variants Microchip (-semi)



# Microchip PolarFire SoC Icicle kit (iccl)

- €549 (digikey.de)
- Device MPFS250T-FCVG484EES
- Price N/F
- Host: Yocto Linux on RISC-V
- ✓ (PCle)
- ✓ (MIPI CSI-2)
- X HDMI out
- ✓ 2GB LPDDR4 SDRAM

PCIe, MIPI CSI-2, HDMI and DDR memory across platforms from an Open Source perspective





• Up-to-date project documentation on GitHub (to receive all information of this presentation and more)

#### https://mpsitech.github.io/Laser-Scanner-By-Platform

 Documentation of Whiznium, Open Source tool for model-based source code generation covering the "embedded full-stack" (gives instructions to set up the tools themselves but also SoC-related things like Yocto/Buildroot)

https://github.com/mpsitech/The-Whiznium-Documentation

• The Whiznium Starter Kit source code (Linux daemon)

https://github.com/mpsitech/wzsk-Whiznium-StarterKit

• The Whiznium Starter Kit Device source code (RTL for all platforms)

https://github.com/mpsitech/wskd-Whiznium-StarterKit-Device



# FPGA(-SoC) project variants

- Expected completion by mid-September 2023
- Weekly updates
- Advice to investigate once "code" columns show "published" for a given platform

○ ○ S Welcome to the Laser Scanner × +

 $\leftrightarrow$   $\rightarrow$   $\bigcirc$  e mpsitech.github.io/Laser-Scanner-By-Platform/

A Laser Scanner by Platform

Microchip PolarFire SoC Icicle kit (iccl)

Aries Cyclone V SE SoM (mcep) Digilent Nexys Video (nxsv)

Efinix Titanium Ti180 dev kit (tidk) SiLabs UniversalBee dev kit (ubdk)

Avnet ZUBoard 1CG (zudk)

DDR memory

HDMI (out) MIPI CSI-2 PCI Express

PLATFORMS

Toradex Apalis (apalis) Digilent Arty Z7-20 (arty) terasIC Cyclone V GX starter kit (cgsk) Lattice CrossLink-NX eval board (cleb)

#### **Completion Matrices**

#### Platforms

Status for each hardware platform.

Platform	PCB's	VHDL/C code	host code	DDR	HD
Toradex Apalis	ready	not applicable	published	not applicable	not
Digilent Arty Z7-20	ready	ready	ready	ready	rea
terasIC Cyclone V GX starter kit	ready	in progress	in progress	in progress	in p
Lattice CrossLink-NX eval board	ready	ready	ready	not applicable	not
Microchip PolarFire SoC Icicle kit	ready	in progress	ready	not started	not
Aries Cyclone V SE SoM	ready	not started	not started	not started	not
Digilent Nexys Video	ready	in progress	ready	in progress	rea
Efinix Titanium Ti180 dev kit	ready	in progress	in progress	ready	rea
SiLabs UniversalBee dev kit	ready	in progress	ready	not applicable	not
Avnet ZUBoard 1CG	ready	in progress	ready	ready	no

#### Comments

[1] more suitable platform to be found where clock and signals are routed differentially to a connector

#### FPGA Interfaces

Status of the wrappers, aiming to make standard FPGA interfaces as platform-independent as possible. Results are made available in the respective current WhizniumDBE release.

Platform	DDR memory	HDMI (out)	MIPI CSI-2	PCI Express
AMD Gen1 (arty, nxsv)	in progress	in progress	in progress	in progress
AMD Gen2 (zudk)	in progress	not planned	in progress	not planned
Efinix (tidk)	in progress	in progress	in progress	not planned
intelPSG Gen1 (cgsk, mcep)	not started	not started	not started	not started
Lattice (cleb)	not planned	not planned	in progress	in progress
Microchip (iccl)	not started	not planned	not started	not planned



#### Overview | Hardware prerequisites | FPGA integration | Resources

- PCI Express Gen1 (introduced 2003 as replacement for parallel PCI) offers 2.5 Gbps per lane
- Focus on x1, i.e. single data lane, net data rate of 250MB/s

Physical layer:

- Point-to-point communication (here: host-to-FPGA), FPGA is endpoint
- NRZ line code: clock (1.25GHz, 0.4ns eye) is recovered from signal
- Reference clock (100MHz) signal only present during system start-up
- Among those presented here, the most demanding interface, impossible without dedicated silicon
- I<sup>2</sup>C interface not relevant for FPGA implementation

Configuration / transaction layer:

- Configuration ("capabilities") read by host from FPGA during system start-up
- Focus on "Memory Write" / "Memory Read" only with 1k payload size per transaction



#### Overview | Hardware prerequisites | FPGA integration | Resources

- Max. allowed length of PCIe v1.0 lane is 38 cm (15 inch)
- For practical purposes ubuntu Linux host with Mini-PCIe breakout beats workstation / riser card: example intel NUC 11 i3 with converter M.2 Key A+E to mini PCIe slot (DeLOCK 62848) and a mini PCIe breakout (DeLOCK 41370)
- Some Embedded Linux hosts (e.g. Toradex Apalis) also feature mini PCIe slots but this adds complication (Yocto) to an already complicated topic





## Overview | Hardware prerequisites | FPGA integration | Resources

#### Nice-to-have block structure

Connection to hardware:

- Reference clock (diff.)
- RXD (diff.)
- TXD (diff.)

Connection to RTL:

AXI stream data out

AXI stream data in (both: 62.5MHz @ /32, 31.25MHz @ /64)

- Register-based configuration/status



FPGA boundary



## Overview | Hardware prerequisites | FPGA integration | Resources

• AMD: AXIS master for RX, AXIS slave tor TX (as seen from FPGA)

	Re-customize IP	
Series Integrated Block for PCI Express (3.	3)	- 🍐
Documentation 📄 IP Location C Switch to Defaults		
Show disabled ports	Component Name price_x1         Rait:       UP         Node       Basic:         Device Port Type:       Cites Endpoint device:       Xillins Development Board         Device Port Type:       Cites Endpoint device:       Xillins Development Board       None         PCE Bick Location       VOID       Silicen Revision       GES and Production       V         Number of Lanes       Maximum Link Speed       Silicen Revision       GES       ADI Interface Width         Prequency UM40       62.5       XI Interface Width       Here       V       V       Maximum Link Speed       Silicen Revision       Totale       Totale       Totale       Totale       Totale       None       Totaled PROM (Refer PGOS1)       Toutele Configuration       V<	
	ОК	Cancel

• intelPSG: 32 bit Avalon interface



### Overview | Hardware prerequisites | FPGA integration | Resources

• Lattice: AXIS master for RX, AXIS slave tor TX, APB config with 32 bit address mapped registers





## Overview | Hardware prerequisites | FPGA integration | Resources

#### • Linux host perspective: lsusb

```
mpsitech@chip:~$ sudo lspci -v
00:00.0 Host bridge: Intel Corporation Device 9a04 (rev 01)
             DeviceName: Onboard - Other
             Subsystem: Intel Corporation Device 3002
             Flags: bus master, fast devsel, latency 0
             Capabilities: [e0] Vendor Specific Information: Len=14 <?>
. . .
00:02.0 VGA compatible controller: Intel Corporation Device 9a78 (rev 01) (prog-if 00 [VGA controller])
. . .
01:00.0 Serial controller: MosChip Semiconductor Technology Ltd. 4-Port PCIe Serial Adapter (prog-if 02 [16550])
             Subsystem: Asix Electronics Corporation (Wrong ID) 4-Port PCIe Serial Adapter
             Flags: fast devsel, IRQ 16
             I/O ports at 3030 [size=8]
             Memory at 4fa07000 (32-bit, non-prefetchable) [size=4K]
             Memory at 4fa06000 (32-bit, non-prefetchable) [size=4K]
. . .
```

- PCIe typically only makes sense combined with DMA and interrupts
- the Linux kernel's Userspace I/O has corresponding features, already integrated with PCIe



### Overview | Hardware prerequisites | FPGA integration | Resources

#### A. Vendor-provided IP documentation, user guides, product guides

- 1. AMD: 7 Series FPGAs Integrated Block for PCI Express (UG477) [IP encrypted, included in free license]
- 2. intelPSG: Cyclone V Hard IP for PCI Express (UG-01110) [IP encrypted, included in free license]
- 3. Lattice: PCIe X1 IP Core Lattice Radiant Software (IPUG-02091) [IP encrypted, not free (4-hour expiry)]
- 4. Microchip: PolarFire SoC FPGA PCI Express (UG0920) PolarFire FPGA and PolarFire SoC FPGA PCI Express (DS50003188B)
- B. Publicly available use cases / examples
- C. Relevant work by Open Source community
  - 1. <u>https://github.com/alexforencich/verilog-pcie</u>



#### **Overview** | Hardware prerequisites | FPGA integration | Resources

- MIPI CSI-2 is the (only remaining?) standard for megapixel camera sensors
- Market dominance of SONY IMX range, from ArduCam's to internal iPhone cameras

Physical layer:

- Up to four lanes together with one clock lane, matching via FPGA-internal delay line
- Time-division between "low-power" (LP) 1.2V CMOS and "high-speed" (HS) LVDS modes
- DDR protocol with e.g. 1600 Mbps per lane @800 MHz

#### Protocol:

• "LP announces what comes in next HS chunk", bit packing and distribution across lanes (RAW/YUV/...) specified by supplier

#### Configuration:

- I<sup>2</sup>C interface configures everything: number of lanes, resolution, image format, frame rate, exposure time, ...
- without access to datasheet mostly stuck with default (workarounds: Linux kernel sources, contact SONY / FRAMOS)



#### Raspberry Pi Cam lineup

Two-lane:

V2: 8MP IMX219

Optionally four-lane:

- HQ: 12.3MP IMX477
- V3: 11.9MP IMX708

### Overview | Hardware prerequisites | FPGA integration | Resources

- Current generation FPGA(-SoC)'s feature native MIPI support
- AMD and intelPSG each published an application note showing how to split LP / HS externally with passives



(source: intelPSG AN754)

• ... but this is recommended for < 800 Mbps only, which is exceeded even by most ArduCam's



## Overview | Hardware prerequisites | FPGA integration | Resources

#### Nice-to-have block structure

Connection to hardware:

- clock (diff.)
- N lane RX (diff.)
- I<sup>2</sup>C

Connection to RTL:

- AXI stream RGB/YUV data out (e.g. 5 MP x 10 bit x 60 fps = 3000 Mbps => 93.75 MHz @ /32, < 50 MHz @ /64)</li>
- {get/set}RegVal configuration



FPGA boundary



#### Overview | Hardware prerequisites | FPGA integration | Resources

• AMD: per-lane 1:8 gearing vs. AXIS master fixed 16 bit

Customize IP 8			8 Re-customize IP				
MIPI D-PHY (4.3)			4	MIPI CSI-2 Rx Subsystem (5.2)		A	
ODcumentation 🕞 IP Location C Switch to Defaults				Occumentation 🗇 IP Location C Switch to Defaults			
<pre>Show disabled ports</pre>	Component Name mpl_dphy_0 Core Parameters NOELAY_GROUP Name D-PHY Lanes Le Rela (Mpp.) Data Flow Esc Ck (Msb) LYX Penod (ns) D-PHY RX ULPS WAREUP counter fo Resource optimization presets Control and Debug Resource optimization presets Control and Debug D-PHY RX ULP Register (F Protocol Watchdog Timery Escape Timeout (ns) Calibration Mode @ NoNE _ PRED_AUTO DELAY Tap Value 1	ipi_dphy_idly_group 0           4           900         0           900         180 - 1500           asheet         20.000         180 - 20.0           50         0         150 - 100           or 1 ms time         None         •           rers/Registers         95541         •           12500         1800 - 25501         11 - 31	ancel	Show disabled ports         iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii	Component Name impiny w2_1188 Configuration Application Example Design Subsystem Options Pluel Format RAW10 V Serial Data Lanes 2 V Onclude Video Format Bridge (VR8) Support CSI Spec V2_0 DPHY Options IDDELAY_GROUP Name Inc.esizor, ub/, group 8 Une Rate (Mps) 1188 (0.1500) Une rate supported by Device Datasheet DPHY Register Interface Other Register Interface Elibration Mode (0.1512 Options CSI2 Controller Register Interface Embedded non-image Interface Embedded non-image Interface Embedded non-image Interface Embedded non-image Interface Embedded non-image Interface Miler User Defined data types Une Buffer Depth 2048 V VIB Options Allowed Vic All V Puels Per Clock 1 V		
		OK	ancel		OK	Cancel	



### Overview | Hardware prerequisites | FPGA integration | Resources

• Efinix: full decoding to 64 bit pixel data plus sync

	IP Con	figuration		🛛
Module Name:				0
				0
efx_csi2_rx_top 1.3	General Deliverables Summary			
	Data Lanes	4 •		<b>^</b>
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	IP core frequency(mhz)	100		
reset_pixel_n $\longrightarrow$ vsync_vc7 clk_pixel $\longrightarrow$ vsync_vc14	Pixel data fifo depth size	1024 *		
$axi_rready \longrightarrow $ vsync_vcl3 $axi_arvalid \longrightarrow $ vsync_vcl1 $axi_araddr[5:0] \longrightarrow $ vsync_vcl0	Image Frame Mode	GENERIC +		
axi_bready → vsync_vc8 axi_wvalid → vsync_vc6 axi_wdata[31:0] → vsync_vc4	Enable extra bits on Virtual Channel	Disable 👻		
axi_clk → vsync_vc0 axi_reset_n → vsync_vc5 axi_awaddr[5:0] → irq	Enable pipeline state for RXStopState signal	8		
axi_awvalid RxUlpsClkNvt[0:0] xUlpsActiveClkNvt[0:0] pixel_data[63:0] pixel_data[63:0] pixel_data[63:0]	Number of asynchronous register stages	2		_
$\begin{array}{cccc} RxClkEsc[3:0] & \longrightarrow \\ RxErrEsc[3:0] & \longrightarrow \\ RxErrControl[3:0] & \longrightarrow \\ word_count[15:0] \end{array}$	tiNIT(ns)	100000		
$\begin{array}{cccc} RxErrSotSyncHS[3:0] &\longrightarrow & vcx[1:0] \\ RxUlpsEsc[3:0] &\longrightarrow & vc[1:0] \\ RxUlpsActiveNot[3:0] &\longrightarrow & hsync_vc3 \end{array}$	Pack Type 40	Enable *		
RxSkewCalHS[3:0] BxStapState[3:0] byvc vc8	Pack Type 48	Enable 👻		*
RxSyncHS[3:0] $\longrightarrow$ hsync_vc12			Reset defau	lt settings
			Show Confirmation Box Generate	Close



### Overview | Hardware prerequisites | FPGA integration | Resources

• Lattice: 4:64 gearing vs. AXIS master

agram mipidphy		Configure mipidphy:	
	^	Property	Value
mir	aidaby	▼ Configuration	
1111		Interface Type	Receive
-hs data des en i	clk_byte_o—	MIPI Interface Application	CSI-2
-hs rx ck en i	ck_n_io —	DPHY Module Type	Hard MIPI DPHY
hs rx data en i	clk_p_io —	HARD_IP DPHY Mode	NO CIL
-Immi clk i	data_n_io[3:0]	DPHY PLL Mode	External
Immi_offset_i[4:0]	data_p_io[3:0]	DPHY Clock Mode	Continuous
-Immi_request_i	hs_rx_data_o[63:0]	Interface Data Rate (Mbps) [80 - 2500]	1184
_lmmi_resetn_i	is_rx_data_sync_o[3:0]	Gearing Ratio	1:16
- Immi_wdata_i[3:0]	immi_rdata_o[3:0]	Bus Width	4
- lmmi_wr_rdn_i		Reference Clock Frequency (MHz) [24 - 200]	24
lp_rx_en_i		Interface Clock Frequency (MHz) [40 - 1250]	592
-pd_dphy_i			
-pl_lock_i	lo rx data n o[3:0]		
-sync_clk_i	p rx data p o[3:0]		
-sync_rst_i	ready_o-		
min	i dabu		
mip	Labuy		

ram mipirx		Configure IP		
-	•	Property	Value	<b>^</b>
mi	pirx	▼ Receiver		
	avis mtdata o[63:0]	RX Interface	CSI-2	
	axis mtvalid o	D-PHY RX IP	Hard D-PHY	
	bd_o[31:0]	Number of D-PHY Data Lanes	4	
	clk byte bs o	RX Gear	8	
axis_stready_i	clk_byte_ns_o	CIL Bypass		
clk_byte_fr_i	cik_byte_o	Enable LMMI Interface		
pd_dphy_i		Enable AXI4-Stream Interface	<b>V</b>	
ref_dt_i[5:0]	d n io[3:0]	Enable Deskew Calibration Detection		
reset_byte_fr_n_i	d_n_io[3:0]	▼ Clock		
reset_byte_n_i		RX Line Rate (Mbps) [80 - 1500]	1184	
reset_n_i	Ins_sync_0	D-PHY Clock Frequency (MHz) [40 - 1250]		
tx_rdy_i	ip_av_en_o	D-PHY Clock Mode	Continuous	
	Ip_a_rx_n_o[3:0]	Byte Clock Frequency (MHz) [10 - 187]	148	
	ip_a_rx_p_o[3:0]	Sync Clock Frequency (MHz) [60 - 200]	60	
	ip_en_o	<ul> <li>Timing Parameter</li> </ul>		
	sp_en_o	Customize Data Settle Cycle		
dph	iy_rx	Data Settle Cycle [11 - 22]	14	
		Customize CIL Data Settle		-



### Overview | Hardware prerequisites | FPGA integration | Resources

#### A. Vendor-provided IP documentation, user guides, product guides

- 1. AMD: 7 Series FPGAs SelectIO Resources (UG471) [included in free license] MIPI CSI-2 Receiver Subsystem (PG232) [included in free license]
- 2. Efinix: Ti180 Data Sheet (DSTi180) chapter on MIPI D-PHY [IP encrypted, included in free license]
- 3. intelPSG: to be explored
- 4. Lattice: MIPI D-PHY Module Lattice Radiant Software (IPUG-2061) [IP encrypted, included in free license] CSI-2/DSI D-PHY Rx IP Core - Lattice Radiant Software (IPUG-02081) [IP encrypted, not free (4-hour expiry)]
- 5. Microchip: PolarFire FPGA and PolarFire SoC FPGA User I/O (DS60001727C) MIPI CSI-2 Receiver Decoder For PolarFire (UG0806)
- B. Publicly available use cases / examples
- C. Relevant work by Open Source community
  - 1. <u>https://github.com/gatecat/CSI2Rx</u>
  - 2. <u>https://github.com/hellgate202/csi2\_rx</u> (careful, GPLv3)



### Overview | Hardware prerequisites | FPGA integration | Resources

• HDMI is one of the major standards to transmit image data to screens / projectors

Physical layer:

- Three data lanes (typ. red/green/blue) together with one clock lane running at a 10<sup>th</sup> of the data speed
- Clock range 25 MHz to 165 MHz (full-HD @ 60Hz)
- TMDS packs 8 bits into 10 bits, i.e. @ 165 MHz, each lane carries 1.65 \* 8/10 Gbps = 1.32 Gbps

Protocol:

vsync and hsync are embedded into the "blue" TMDS lane



### Overview | Hardware prerequisites | FPGA integration | Resources

- LVDS transmitters can drive TMDS lines directly (probably not meeting power levels of official spec)
- Standalone (preferred Digilent arty / nxsv): amplifier e.g. TMDS141
- External processor (preferred Efinix tidk / terASIC cgsk): parallel-to-TMDS converter e.g. ADV7511



## Overview | Hardware prerequisites | FPGA integration | Resources

#### Nice-to-have block structure

Connection to hardware (standalone):

- TMDS clock (diff.)
- 3x TMDS data (diff.)

#### Connection to hardware (external processor):

- clock
- 3x 8 .. 12 RGB data
- vsync, hsync

#### Connection to RTL:

- AXI stream 3x 8 .. 12 RGB data
- vsync, hsync



FPGA boundary



## Overview | Hardware prerequisites | FPGA integration | Resources

- A. Vendor-provided IP documentation, user guides, product guides
- B. Publicly available use cases / examples
  - 1. https://www.fpga4fun.com/HDMI.html, https://www.youtube.com/watch?v=sMOZxOCfkBU
  - 2. https://digilent.com/reference/learn/programmable-logic/tutorials/arty-z7-hdmi-demo/start
  - 3. Efinix\_Ti180M484\_oob out-of-the-box design (requires login)
- C. Relevant work by Open Source community



### Overview | FPGA integration | Resources

• (LP)DDR{2-6} is the interfacing standard for large dynamic RAM IC's

Physical layer:

- Column / row address inputs (start address)
- Single-ended bidirectional data lanes (typ. 8/16/32 bits) with DDR sampling
- One differential pair of synchronization signals per byte
- DDR protocol e.g. DDR3-800 @400MHz

Protocol:

- Sophisticated "training" procedure in PHY to equalize wire lengths
- Then burst transfer with IC-specific delay from column / row address input to data



(source: MT42L16M32D1 data sheet)



### Overview | FPGA integration | Resources

#### Nice-to-have block structure

Connection to hardware:

- Adressing
- DQS /1, /2, ... (diff.)
- DQ /8, /16, ... (bi-directional)

Connection to RTL:

 Set of channels with "reduced" AXI4 providing start address, configured for burst transfers



FPGA boundary



#### Overview | FPGA integration | Resources

- AMD (non-SoC): AXI4
- Efinix: AXI4

ule Name:				D
_ddr3_soft_controller 1.8	General MR0 MR1 MR	2 MR3 Timing Arbiter Deliverables	Summary	
i_dqs_hi[1:0]	Simulation	no 👻		
$i_dqs_lo[1:0] \longrightarrow shift[2:0]$				
$i_dqs_n_hi[1:0] \longrightarrow shift_sel[4:0]$	DDR3 Interface	AXIA		
i_dqs_n_lo[1:0] → shift_ena	DDRS menace			
i dg lo[15:0] $\longrightarrow$ cal_done				
wr data[127:0]  cal fail log[6:0]	Read Enable Pipeline	8		
wr_datamask[15:0] $\longrightarrow$ cal_shift_val[2:0]				
$axi_aid[7:0] \longrightarrow o_dm_hi[1:0]$	DDR3 DATA RATE	800D -		
axi_aaddr[31:0] $\longrightarrow$ o_dm_lo[1:0]				
$axi_alen[7:0] \longrightarrow o_dqs_hi[1:0]$		-		
axi_asize[2:0] $\longrightarrow$ o_dqs_lo[1:0]	DDR DQ Width	16 *		
axi_abursq1:0] axi_abursq1:0] axi_abursq1:0]				
axi avalid $\longrightarrow$ o dgs oe[1:0]	DDR Frequency	400 🗘		
axi_atype				
axi_wid[7:0] $\longrightarrow$ o_dq_hi[15:0]	Column	10		
axi_wdata[127:0] $\longrightarrow$ o_dq_lo[15:0]				
axi_wstrb[15:0] $\longrightarrow$ o_dq_oe[15:0]				
axi_wiast axi_aready	Row			
axi rready axi wready				
axi_bready	Bank			
axi_rdata[127:0]				
axi_rlast	Bank	1		
axi_rvalid     axi_rvalid				
axi_rresp[1:0]			Reset default se	attin

• intelPSG (non-SoC): Avalon



#### Overview | FPGA integration | Resources

- The SoC case is special in that the physical memory is shared between CPU complex and FPGA
- Allowable AXI4 burst length may be reduced (e.g. max. 16 beats @128 bit for MPSoC)
- DDR controller configuration typically register mapped in common address space
- Configuration typically through first-stage boot loader
- May need to declare "reserved" section e.g. in Linux device tree



(source: Xilinx UG1085)



# DDR Memory Overview | FPGA integration | Resources

#### A. Vendor-provided IP documentation, user guides, product guides

- AMD: Zynq-7000 SoC and 7 Series Devices Memory Interface Solutions (UG586) [IP encrypted, included in free license] Zynq-7000 All Programmable SoC TRM (UG585) chapter 10 Zynq UltraScale+ Device TRM (UG1085) chapter 17
- 2. Efinix: Titanium DDR DRAM Block (UG-DDR-DRAM-TI)
- 3. intelPSG: DDR2 SDRAM Controller with UniPHY [IP unencrypted, included in free license]
- 4. Microchip: PolarFire SoC FPGA DDR Memory Controller (UG0906)
- B. Publicly available use cases / examples
- C. Relevant work by Open Source community



# Remarks

## High-speed connectors | Cross-vendor development

Formerly:

- PMOD
- FMC by Xilinx (also adopted by Efinix and Lattice)
- HSMC by Altera

New (low pin count but high-speed):

- SYZYGY, SYZYGY TXR by AMD / OpalKelly (also adopted by Efinix / sort of)
- CRUVI by intelPSG / Trenz



# Remarks

### High-speed connectors | Cross-vendor development

- Spend most time outside of vendor IDE's
- VS Code with standalone GHDL or TerosHDL works really well, beyond syntax checking
- Use vendor IDE's to extract IP wrappers, work with black boxes as long as possible
- Back to vendor IDE's for actual synthesis-to-bitstream workflow





 Coverage of the "Embedded Full Stack"

- WhizniumDBE ("Device Builder's Edition") for FPGA / MCU level and its host access libraries (primary languages: C, VHDL)
- WhizniumSBE ("Service Builder's Edition") for Embedded Linux and "outside world" levels (primary languages: C++, HTML)

PCIe, MIPI CSI-2, HDMI and DDR memory across platforms from an Open Source perspective



# Model-based software design with Whiznium

#### **Overview |** Concepts | Examples

- Successive model composition within an SQL database using import (I) and generation (G) steps
- Output of source code trees only thereafter
- Text-based model files ("diffable")

#### WhizniumDBE (Device Builder's Edition)

- Modular structure (I)
- Command set and buffer transfers (I)
- Data flows and algorithms (I)
- Fine structure (G)
- Custom fine structure (I)
- Finalization (G)

#### WhizniumSBE (Service Builder's Edition)

- Deployment information (I)
- Global features (I)
- Database structure (I)
- Basic user interface structure (I)
- Import/export structure (I)
- Operation pack structure (I)
- Custom jobs (I)
- User interface (G)
- Custom user interface features (I)
- Job tree (G)
- Custom job tree features (I)
- Finalization (G)



- Whiznium is Open Source; the generated code is subject to no license restrictions
- Whiznium generates well-organized, human-readable source code trees which can be synthesized / compiled "outof-the-box"
- Manual modifications are enabled through the concept of "insertion points"
- Upon source code iteration (e.g. following model extension) manual modifications are carried over to the next version
- Generated code relies on few, well-proven external libraries, all of which are Open Source. Standards are strictly followed
- WhizniumDBE features parametrized "module templates". Besides corresponding VHDL files, template-specific intervention in the WhizniumDBE master database through C++ code is possible
- WhizniumSBE features parametrized "capability templates". Also here, template-specific intervention in the WhizniumSBE master database through C++ code is possible



- For the ability to map and manipulate modular and fine struture of RTL projects, WhizniumDBE is pre-destined for providing the cross-vendor wrappers of the interfaces discussed here
- Many thoroughly tested templates already included: UART/SPI/I<sup>2</sup>C, dual-port RAM generator, ...
- Examples for WhizniumDBE + WhizniumSBE interplay
  - Terminal in web-based UI to send commands to / bulk data to/from FPGA
  - Analyzer in web-based UI to track live finite state machine coverage



- "Signature" WhizniumDBE template: host interface, providing RTL and C++ code for host <-> FPGA communication
- From ...

fpga	lifcl-40-9bg400c	cleb	Lattice CrossLink-NX Evalua	true	radiant
	ImelMModule.sreflxVBasetype	hsrefSupRefWdbeMModule	srefTplRefWdbeMModule	sref	Comment
	top		top_v1_0	top	
		ImeIAMModulePar.x1SrefKKey	Val		
		clkIntNotExt	true		
		extresetNNotP	true		
		fExtclk	50000		
		ImeIAMModulePar.end			
		ImeIMGeneric.sref	Defval		
		fMclk	50000		
		ImelMGeneric.end			
	ehostif	top	hostif Easy v1 0	hostif	connection t
		ImeIAMModulePar.x1SrefKKey	Val		
		phytype	uart		
		fSclk	115200		
		ImeIAMModulePar.end			

	ImelMUnit.sreflxVB	srefSilRefWdbeMUnit	sref	Title	Easy	srefKToolch
	fpga	xc7z020-1clg400	arty	Digilent Arty Z7	true	vivado
		ImelMModule.srefixVBasetype	hsrefSupRefWdbeMModule	srefTplRefWdbeMModule	sref	Comment
		top		top_v1_0	top	
			ImelAMModulePar.x1SrefKKey	Val		
			clkIntNotExt	false		
			extresetNNotP	true		
- 1/5 -			extclkDiffNotSng	false		
- v3			fExtclk	100000		
			ImelAMModulePar.end			
			ImelMGeneric.sref	Defval		
			fMclk	50000		
			ImelMGeneric.end			
		ehostif	top	hostif_Easy_v1_0	hostif	
			ImelAMModulePar.x1SrefKKey	Val		
			phytype	axi		
			wA	32		
			wD	32		
			ImelAMModulePar.end			



# Model-based software design with Whiznium

#### Overview | Concepts | Examples





# Conclusion

- Success in working with the new low-cost FPGA interfaces still is hard earned
  - Non-standardized hardware / connectors are one obstacle
  - Complexity / non-open IP of vendors are another obstacle
- Unifying concepts such as used in Linux kernel + device tree are missing in the FPGA world
- Model-based source code generation can help
  - WhizniumDBE comes with tested cross-vendor modules and will also include interface wrappers
  - WhizniumDBE maintains a coarse-to-fine project model in a database and is user-extensible (by means of C++ code, e.g. for frequently used IP)



# Thank You! Questions?

#### Also, feel free to connect.

- <u>https://www.linkedin.com/in/wirthmua</u>
- <u>https://github.com/mpsitech</u>

Alexander Wirthmüller Founder & Director

Phone: +49 (89) 4524 3826 Mobile: +49 (175) 918 5480 E-Mail: aw@mpsitech.com

MPSI Technologies GmbH Agnes-Pockels-Bogen 1 80992 Munich Germany www.mpsitech.com



