

ADM-VA600 Board File

Alpha Data Parallel Systems Ltd.

Document number: AD-AN-0142

Document revision: v1.0 / December 15, 2023

1 Introduction

Xilinx is now a part of AMD

The "Xilinx" trademark may be used in this document in order to avoid confusion when referring to Web pages, documentation and software that predates the phasing out of the "Xilinx" brand following its acquisition by AMD Inc. Any references to Xilinx should be interpreted as references to AMD.

A board file is a high-level description of the major features of the ADM-VA600, which can be used with Vivado's IP Integrator tool. The purpose of the board file is to streamline the process of creating a Block Diagram (BD) design, and:

- Provides *presets* for certain IPs that are frequently used in designs for the ADM-VA600.
- Defines key *board interfaces* such as differential clocks, DDR4 SDRAM interfaces etc.
- Defines the pinout and certain physical constraints for the Adaptive SoC in the ADM-VA600, so that the user need not provide DIFF_TERM, IOSTANDARD, PACKAGE_PIN etc. constraints for the board interfaces defined by the board file.

1.1 Board files and order codes

At time of writing, only a single board file exists for the ADM-VA600, which applies to the ADM-VA600/DEV order code. Should other order codes be created in future, Alpha Data may publish additional board files as needed.

Order code(s)	Board file VLNV	Board features
ADM-VA600/DEV	alpha-data.com:admva600_dev:part0:*	Not space-deployable XCVC1902-1MSIVSVA2197 2 x 8 GiB memory @ DDR4-2133P

Table 1: Available ADM-VA600 files by order code

"Order code(s)" are the order codes for which the board file applies. A particular board file might cover several order codes because certain ordering options are considered "cosmetic" as far as AMD tools such as Vivado, Vitis and Petalinux are concerned. An example of such a cosmetic option might be a conformal coating option.

"Board file VLNV" is the string that Vivado uses to identify a particular board file. A board file VLNV may be passed to various Vivado Tcl commands; for example get_board_parts.



2 Obtaining the board file

The recommended way to obtain the ADM-VA600 board file is to download it from within Vivado as described in AMD UG994.

To download the board file, first open the Vivado Store dialog by selecting $Tools \rightarrow Vivado Store...$ from the Vivado main menu. The following graphic shows the steps for downloading the ADM-VA600 board file within the Vivado Store dialog.



Figure 1: Steps for downloading the ADM-VA600 board file

If successful, a green tick appears next to the board as shown in Figure 2 below:



🍌 Vivado Store		×
Welcome to Vivado Store. You can browse and search the available	applications and	install to your local drive.
<u>T</u> cl Apps Boards Example Designs		<u>G</u> o to Git
$\mathbf{Q} \underbrace{\mathbf{X}}_{\mathbf{x}} \Leftrightarrow \underbrace{\mathbf{Y}}_{\mathbf{x}} \ominus \underbrace{\mathbf{Y}}_{\mathbf{x}} \underbrace{\mathbf{Y}}_{\mathbf{x}}$	Details	
Q- > iWave Systems Technologies > Xilinx > Trenz Electronic GmbH > TUL > Opal Kelly > Leopardimaging > Digilent Inc > Avnet < Alpha Data < Digilent Single Part () ADM-PA100/2MS () ADM-PA101/2MS	Name: Description: Revision: URL: Company:	ADM-VA600/DEV ADM-VA600/DEV VPX Reconfigurable Computing Card 1.0 (Installed) Revision History https://github.com/Xilinx/XilinxBoardStore/tree/2022.2/boards/AlphaData/admva600_dev/1.0 Alpha Data
Refresh Catalog was last updated on 07/25/2023 12:54:56 PN	И	Close

Figure 2: After successfully downloading the ADM-VA600 board file

The Vivado Store dialog can now be closed by clicking on the *Close* button. The ADM-VA600/DEV board part will be available when creating new Vivado projects.

3 Board file features

This section provides a brief overview of the IP presets and board interfaces defined by the ADM-VA600 board file.

3.1 CIPS Block Automation

The board file defines a preset for Control Interfaces & Processing System (CIPS) IP that is used by Vivado's Block Automation feature. To use this feature, create a new CIPS IP instance in a Vivado Block Diagram (BD) design. Then right-click on the CIPS IP instance and select *Run Block Automation...* as in Figure 3:



Diagram			? _	οл×
$\textcircled{\begin{tabular}{ c c c c } \hline \begin{tabular}{ c c } \hline \$	r	★ C 의 E Default	View	~ ¢
* Designer Assistance available. Run Block Automation				
ſ		Block Properties	Ctrl+E	
	▲	Highlight	•	
		Unhighlight		
		Show Resources Estimation		
	×	Delete	Delete	
		Сору	Ctrl+C	
versal_cip	10	Paste	Ctrl+V	
	Q,	Search	Ctrl+F	
	133	Select All	Ctrl+A	
Control Interfaces & Pr	+	Add IP	Ctrl+I	
control, interfaces et ri		Add Module		
	*	Run Block Automation		
	۶	Customize Block		[
		IP Documentation	►	
		Orientation	►	
		Pinning	Þ	
	ľ	Validate Design	F6	
		Create Hierarchy		

Figure 3: Running Block Automation on the CIPS IP instance

The Block Automation dialog for the CIPS IP has various options which can be set as per applicament requirements, for the most part. However, in order to configure the PS/PMC module correctly for the ADM-VA600, set *Apply Board Preset* to *Yes* as in Figure 4:



omatically make connections in your design	i by checking the boxes of the blocks to connect. Select a block on the left to display its configuration options on the right.				
a ≚ ≑	Description				
 All Automation (1 out of 1 selected) # versal_cips_0 	The Control, Interface, and Processing System block automation wizard assists with the generation of a NOC switch block. It will also assist with the connections to Memory Controllers and programmable fabric interface ports. Instance: /versal_cips_0				
	Options				
	Design Flow Full System V				
	Configurations				
	Apply Board Preset Yes 🗸				
	Debug Configuration JTAG 🗸				
	PL Clocks 2 V				
	PL Resets 1 🗸				
	Memory Controller				
	Type DDR V DDR Number 2 V				
	NoC				
	Configure NoC Add new AXI NoC 🗸				

Figure 4: Typical CIPS Block Automation settings

NOTE: If the *Apply Board Preset* option is *Yes*, Block Automation applies a particular CIPS IP preset named ps_pmc_fixed_io. See Section 3.2 for information about CIPS IP presets.

The result of the above Block Automation settings is shown in Figure 5:



Figure 5: After running Block Automation

The above is a good starting point for a design that uses the Adaptive SoC's APU and/or RPU embedded processors, since it includes DDR4 SDRAM controllers.



3.2 CIPS IP presets

The board file defines several presets for the CIPS IP which offer a convenient way to configure the CIPS IP without using Block Automation. They are summarized in Table 2 below:

Preset name	PS I/O interfaces	Boot modes	IPI
<pre>ps_pmc_fixed_io</pre>	Configured	JTAG, QSPI, SD card (2.0)	Not configured
<pre>ps_pmc_fixed_io_minimal</pre>	Not configured	JTAG, QSPI, SD card (2.0)	Not configured
<pre>ps_pmc_fixed_io_linux</pre>	Configured	JTAG, QSPI, SD card (2.0)	Configured
ps_pmc_fixed_io_smap32	Configured	JTAG, QSPI, SelectMap (32-bit)	Not configured
<pre>ps_pmc_fixed_io_smap32_minimal</pre>	Not configured	JTAG, QSPI, SelectMap (32-bit)	Not configured
<pre>ps_pmc_fixed_io_smap32_linux</pre>	Configured	JTAG, QSPI, SelectMap (32-bit)	Configured

Table 2: Available CIPS IP presets

In all presets, the PS reference clock is set to 50 MHz, the PS I/O bank voltages are correctly set and UART0 is enabled (for debugging boot problems).

NOTE: The ps_pmc_fixed_io preset is used by Block Automation if the *Apply Board Preset* option is *Yes* (see Figure 4.

The suffices that make up the names of the presets have the following meanings:

- Presets that include a suffix _minimal only configure the PS reference clock, UART0 and boot modes, and omit configuring the rest of the PS I/O interfaces supported in the ADM-VA600.
- Presets that include a suffix _linux configure Inter-Processor Interrupts (IPI) so that the APU and RPU can communicate with the PMC. This is a requirement for running Petalinux in the APU.
- Presets that include a suffix _smap32 replace the SD card (2.0) boot mode with the SelectMap (32-bit) boot mode.

A CIPS IP preset is applied by customizing the CIPS IP instance and selecting it for **Board Interface**, as shown in Figure 6:



Documentation 🌣 Presets 🕒 I	P Location			
mponent Name versal_cips_0				
Design Flow Full System	~			
Board Interface	ps pmc fixed io linux	~		
Boot Configuration	Custom	~	c	
Clock Settings	Custom	~	С	
Connectivity to MC via NoC	Custom	~	C	
I/O Peripherals	Custom	~	C	
Debug	Custom	~	C	
Device Integrity	Custom	~	C	
PS PL Connectivity	Custom	~	C	

Figure 6: Applying a CIPS IP preset

NOTE: It is possible to change the preset after running Block Automation. Thus, a convenient way to generate a Block Diagram design capable of running Petalinux in the APU is to run Block Automation (see Section 3.1) and then change **Board Interface** to one of the _linux presets.

3.3 DDR4 SDRAM interfaces

In Versal architecture, memory controllers are built into the silicon and are considered part of the Network on Chip (NoC). The ADM-VA600 has two banks of DDR4 SDRAM and the board file accordingly provides pinouts and IP presets for them. The DDR4 SDRAM-related board interfaces defined in the ADM-VA600 board file are summarized in Table 3:

Interface name	Description
ddr4_bank0	Bundle of signals compromising the physical interface to DDR4 SDRAM bank 0.
ddr4_bank1	Bundle of signals compromising the physical interface to DDR4 SDRAM bank 1.
ddr4_bank0_sys_clk	Differential reference clock for memory controller for DDR4 SDRAM bank 0.
ddr4_bank1_sys_clk	Differential reference clock for memory controller for DDR4 SDRAM bank 1.

Table 3: Available DDR4 SDRAM-related interfaces

The AXI NoC IP is used to implement a DDR4 SDRAM controller which may use DDR4 SDRAM bank 0 only, bank 1 only or both banks interleaved. Figure 7 below shows the interleaved case:



XI NoC (1.0) Documentation 🕞 IP Location								
	Compone	ent Name a	ixi_noc_0					
	Board	General	Inputs	Outputs	Connectivity	QoS	Addres	4 🕨
	Associat	e IP interfac	e with boa	rd interface	•			
	IP Inter	rface			oard Interface			
	CH0_DI	OR4_0		(ddr4 bank0			•
	CH0_DI	CH0_DDR4_1			ddr4 bank1			•
	CH0_DI	DR4_2	(Custom			Ŧ	
	CH0_DI	CH0_DDR4_3			Custom			w
+ SOD_AXI MOD_AXI + + sys_cik0 cup ppp p 5	CH1_DI	CH1_DDR4_0			Custom			*
+ sys_clk1 CH0_DDR4_0 X - aclk0 CH0_DDR4_1 X	CH1_DI	DR4_1	(Custom			Ŧ	
	CH1_DI	DR4_2	(Custom			Ŧ	
	CH1 DI	DR4 3		(Custom			w
	sys_clk0	svs clk0			ddr4 bank0 sys clk			•
	sys_clk1	svs clk1			ddr4 bank1 sys clk			*
	sys_clk2	svs clk2			Custom			v
	svs clk3	sys_clk3			Custom			÷
	Clea	ar Board Par	ameters]				

Figure 7: AXI NoC IP configured for interleaved DDR4 SDRAM banks

The single-bank cases are illustrated in Figure 8a & Figure 8b, for DDR4 SDRAM banks 0 & 1 respectively:

Board	General	Inputs	Output	ts Connectivity	QoS	Addres: < → ≡			
Associa	te IP interfac	ce with boa	rd interfa	ce					
IP Inte	erface			Board Interface					
CH0_D	DR4_0			ddr4 bank0		~			
CH0_D	DR4_1			Custom		•			
CH0_D	DR4_2			Custom		Ŧ			
CH0_D	DR4_3			Custom		Ŧ			
CH1_D	CH1_DDR4_0			Custom •					
CH1_D	DR4_1			Custom					
CH1_D	DR4_2			Custom -					
CH1_D	DR4_3			Custom 👻					
sys_clk	0			ddr4 bank0 sys clk		•			
sys_clk	1			Custom					
sys_clk	2			Custom 👻					
sys_clk	3			Custom		Ŧ			

Board	General	Inputs	Outputs	Connectivity	QoS	Addres	•	
Associat	te IP interfac	e with boa	rd interface					
IP Inte	rface		Bo	ard Interface				
CH0_D	DR4_0		do	dr4 bank1			•	
CH0_D	DR4_1		Cu	ustom			*	
CH0_D	DR4_2		Cu	ustom			Ŧ	
CH0_D	DR4_3		Cu	ustom			v	
CH1_D	DR4_0		Cu	ustom			•	
CH1_D	DR4_1		Cu	Custom				
CH1_D	DR4_2		Cu	Custom				
CH1_D	DR4_3		Cu	Custom				
sys_clk(0		do	dr4 bank1 sys clk			*	
sys_clk1	1		Cu	ustom			٣	
sys_clk2			Cu	Custom				
sys_clk3	3		Cu	ustom			w	

(a) AXI NoC IP configured for DDR4 SDRAM bank 0

(b) AXI NoC IP configured for DDR4 SDRAM bank 1

Figure 8: AXI NoC IP single-bank configurations

Setting the board interfaces as described above is sufficient to correctly configure all of the DDR4 SDRAM-



related properties of the AXI NoC IP such as memory timings and geometry. The pinout(s) for the chosen DDR4 SDRAM bank(s) are defined by the board file, and therefore the designer need not provide constraints files that define DDR4 SDRAM bank pinouts.

However, some DDR4-related settings can still be tweaked by the designer. For example:

- DBI usage for reads
- · DBI usage for writes
- Individual memory timings (e.g. t_{FAW})

3.4 Differential clock interfaces

The board file defines various differential clocks:

- GT Quad reference clocks for all GT Quads used in the ADM-VA600.
- PL (fabric) clocks, connected to clock-capable pins.

Please see the **Board** tab in IP Integrator to see the complete list of differential clocks defined by the board file. These clocks can be used in a Block Diagram design by right-clicking on an item and selecting **Connect Board Component**:

Sources	Design	Signals	Board	×		? _	
Q X	\$	▶ *)	×				
ADM-VA	500/DEV						^
🗸 🚍 Clock	Sources () out of 17 (connected)			
0.0	eneral pu	nose progr	ammahla	clock for	DI		- 11
• F	Cle	Board Com	ponent Pr	operties.	Ctr	I+E	- 11
0+F 0+F	Cle PCle	Connect Bo	ard Comp	onent			- 11
O F	Cle	Auto Conne	ect				- 11
O F	rogramma	ble clock 0	(PL)			_	- 11
O F	rogramma	ble clock 1	(GTY_REF	CLK1_103	3)		- 11
O F	rogramma	ble clock 2	(GTY_REF	CLK1_104	4)		- 11
O F	rogramma	ble clock 3	(GTY_REF	CLK1_105	5)		Ť
O F	rogramma	ble clock 4	(GTY_REF	CLK1_10	5)		
O F	rogramma	ble clock 5	(GTY_REF	CLK1_200	5)		
O F	rogramma	ble clock 6	(GTY_REF	CLK1_200))		
O F	rogramma	ble clock 7	(GTY_REF	CLK1_20	1)		
O F	rogramma	ble clock 8	(GTY_REF	CLK0_202	2)		
O F	rogramma	ble clock 9	(GTY_REF	CLK0_203	3)		
O F	rogramma	ble clock 10) (GTY RE	FCLK1 20	04)		\sim

Figure 9: Connecting a board file-defined clock

On doing so, a list of compatible IPs and IP interfaces is offered to the designer:



🝌 Connect Board Component	×
Select an IP block interface for connecting board compo programmable clock for PL'.	nent 'General purpose
Q 素 ≑	
Name	VLNV
SYS_CLK8_IN	^
SYS_CLK9_IN	
+ Clocking Wizard	xilinx.com:ip:clk_wizard:1.0
CLK_IN1_D	
CLK_IN2_D	
+ 1G/10G/25G Switching Ethernet Subsystem	xilinx.com:ip:ethernet_1_10
gt_ref_clk gt_ref_clk	
+ 40G/50G Ethernet Subsystem	xilinx.com:ip:l_ethernet:3.3
gt_ref_clk gt_ref_clk	
V + Universal Serial XGMII Ethernet Subsystem	xilinx.com:ip:usxgmii:1.2 🗸
<	>
	OK Cancel

Figure 10: Choosing an IP and IP interface to use with a board-file defined clock

Vivado then creates an instance of the chosen IP, creates an interface port appropriate for the chosen clock and connects the two together. Figure 11 shows the outcome of this process for a board file-defined fabric_clk:

Diagram	? _ □ ₽	×
$\textcircled{\begin{tabular}{c c c c c } \hline \begin{tabular}{c c c c c c } \hline \begin{tabular}{c c c c c c } \hline \begin{tabular}{c c c c c } \hline \begin{tabular}{c c c c c c } \hline \begin{tabular}{c c c c c c c } \hline \begin{tabular}{c c c c c c c } \hline \begin{tabular}{c c c c c c c c c c c c c c c c c c c $	~	٥
clk_wizard_0 fabric_clk + CLK_IN1_D dk_out1 Clocking Wizard		

Figure 11: Board file-defined clock connected to chosen IP

3.5 General purpose I/O (GPIO) interfaces

The board file defines the following GPIO-style interfaces:

Interface name	Description	
user_led	User-definable LEDs (output only)	
user_switch	User-definable switches (input only)	

Table 4: Available GPIO-style interfaces

As with the differential clock interfaces, the GPIO interfaces can be used in a Block Diagram design by right-



clicking on an item and selecting *Connect Board Component*. For an example of this method of connecting a board interface to an IP instance, see Section 3.4.

Alternatively, a compatible IP may be instantiated (for example, AXI GPIO) and customized to set its IP interfaces. For example, Figure 12 shows an AXI GPIO IP instance configured to use both of the GPIO-style board interfaces:

🝌 Re-customize IP			\times
AXI GPIO (2.0) Documentation 🗁 IP Location		2	2
Show disabled ports	Component Name axi_gpio_0 Board IP Configuration Associate IP interface with board inte	erface	
s_axi_arcsetn GPIO +	IP Interface	Board Interface	1
	GPIO	user led 👻	
	GPIO2	user switch 👻	
	Clear Board Parameters		
	Enable Interrupt		
		OK Cancel	

Figure 12: Configuring IP Interfaces of an AXI GPIO IP instance

3.6 I²C interfaces

The board file defines the following I²C interfaces:

Interface name	Description	
pl_temp_i2c	Interface to temperature sensor.	
vpx_sm_i2c	Interface to VPX system management bus.	

Table 5: Available I²C interfaces

As with the differential clock interfaces, the I²C interfaces can be used in a Block Diagram design by right-clicking on an item and selecting *Connect Board Component*. For an example of this method of connecting a board interface to an IP instance, see Section 3.4.

Alternatively, a compatible IP may be instantiated (for example, AXI GPIO) and customized to set its IP interfaces. For example, Figure 13 shows an AXI IIC IP instance configured to use the pl_temp_i2c board interface:

🝌 Re-customize IP		×
AXI IIC (2.1) Documentation Documentation		4
Show disabled ports	Component Name axi_iic_0 Board IP Configuration	
"(Associate IP interface with board	interface
s_axi_aclk iic2intc_ipt		pl temp i2c •
	Clear Board Parameters	
		OK Cancel

Figure 13: Configuring the IP Interface of an AXI IIC IP instance

4 Board features without board file support

4.1 FMC+ connector

ALPHA DATA

No board interfaces or IP presets are defined for the FMC+ connector because the pinout for it is applicationspecific.

4.2 VPX P1 & P2 connectors

No board interfaces or IP presets are defined for the VPX P1 & P2 connectors because the pinout for them is application-specific.

No board interfaces or IP presets are defined for the CPM PCIe blocks within the CIPS IP because the CIPS IP does not have any board interface properties for them.

5 Related documents

- 1. UG895 Vivado Design Suite User Guide: System-Level Design Entry, AMD Inc. (formerly Xilinx Inc.)
- UG994 Vivado Design Suite User Guide: Designing IP Subsystems Using IP Integrator, AMD Inc. (formerly Xilinx Inc.)
- 3. PG313 LogiCORE IP Product Guide "Versal Adaptive SoC Programmable Network on Chip and Integrated Memory Controller", AMD Inc. (formerly Xilinx Inc.)
- 4. PG352 LogiCORE IP Product Guide "Control Interfaces and Processing System", AMD Inc. (formerly Xilinx Inc.)
- 5. AD-UG-1456 ADM-VA600 User Manual, Alpha Data Parallel Systems Ltd.



Document version history

Document version	Notes
1.0	Initial version.

© 2023 Copyright Alpha Data Parallel Systems Ltd. All rights reserved.

This publication is protected by Copyright Law, with all rights reserved. No part of this publication may be reproduced, in any shape or form, without prior written consent from Alpha Data Parallel Systems Ltd.

Head Office

US Office

Address:	Suite L4A, 160 Dundee Street	Suite 250, 10822 West Toller Drive
	Edinburgh, EH11 1DQ, UK	Deer Creek Technology Center, Littleton, CO 80127
Telephone:	+44 131 558 2600	(303) 954 8768
Fax:	+44 131 558 2700	(866) 820 9956 - toll free
email:	sales@alpha-data.com	sales@alpha-data.com
website:	www.alpha-data.com	www.alpha-data.com

All trademarks are the property of their respective owners.