Automatic Digital Pre-distortion Design Generation for AI Engine

XAPP1391 (v1.0) May 24, 2023

Summary

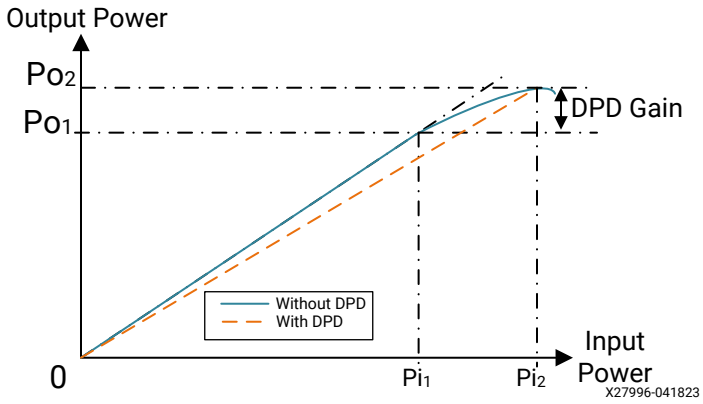
Digital pre-distortion (DPD) technology is widely used in wireless base stations to boost transmit power and mitigate signal distortion incurred in radio-frequency power amplifiers. The generalized memory polynomial model requires DPD to run at three to five times instantaneous signal bandwidth, which becomes challenging for programmable logic as new frequency bands wider than 500 MHz are expected to be available soon for 5G and 6G wireless systems. This application note describes a tool for automatic generation of DPD forward path design on AI Engine without the need of writing a single line of code. The DPD look-up table (LUT) configuration is specified in a plain text file, and the tool generates a complete AI Engine design together with RTL kernels, C drivers, and V++ scripts for hardware validation. This approach turns the AI Engine array into a set of customizable high-performance hardware accelerators for computationally intensive tasks.

Download the reference design files for this application note from the AMD website. For detailed information about the design files, see Reference Design.

Introduction

In 4G long-term evolution (LTE) and 5G new radio (NR) base stations, power amplifiers typically account for 70% of total power consumption in remote radio systems. The following figure shows a diagram of input and output power levels of a typical power amplifier (PA) whose operation region is from 0 to Pi_1 in which the relationship between input and output power is almost linear. Though the output power can increase to Po_2, the intermodulation and other undesirable distortions to the signal resulting from the non-linearity of PA would violate the requirements on adjacent channel interference and spectral emission mask. Digital pre-distortion (DPD) is a technology widely adopted in the industry to improve output power by applying a set of non-linear filters to the digital signal to compensate for the distortions. As a result, the linear region of the PA goes up from Po_1 to Po_2.

Figure 1: **Input and Output Power of Amplifiers**



A widely adopted DPD algorithm is based on the generalized memory polynomial (GMP) architecture proposed in *A generalized memory polynomial model for digital pre-distortion of RF power amplifiers* [4].

Equation 1: **DPD GMP Model**

$$
\begin{aligned}
y_{\mathrm{GMP}}(n) = & \sum_{k\in\mathcal{K}_a}\sum_{l\in\mathcal{L}_a} a_{k,l}x_{n-l}|x(n-l)|^k \\
& + \sum_{k\in\mathcal{K}_b}\sum_{l\in\mathcal{L}_b}\sum_{m\in\mathcal{M}_b} b_{k,l,m}x_{n-l}|x(n-l-m)|^k \\
& + \sum_{k\in\mathcal{K}_c}\sum_{l\in\mathcal{L}_c}\sum_{m\in\mathcal{M}_c} c_{k,l,m}x_{n-l}|x(n-l+m)|^k \\
& = \sum_{m,d} x(n-d)\,\mathbf{f}\left(|x(n-m)|\right)
\end{aligned}
$$

X28080-050523

The last equality combines the GMP terms of the same delay into a polynomial f(|x|) that can be pre-calculated and stored in a LUT. The complexity of DPD is proportional to the number of LUTs, which are indexed by ordered pairs (d, m).

The DPD sample rate should be three to five times that of the instantaneous signal bandwidth. As new frequency bands are being proposed for 5G and beyond, the instantaneous bandwidth is expected to exceed 500 MHz. Thus it becomes a challenge for programable logic (PL) to support a 2 GSPS or higher sample rate. AMD recently announced AI Engine for compute intensive functions such as beamforming, FFT, and DPD. See *AI Engines and Their Applications* (WP506) for more information. Though AI Engine is coded in C++ programming language, there is a learning curve for first-time users. For ease of use, a set of MATLAB® scripts are provided to generate the AI Engine design automatically based on the terms of Volterra series selected by the user. The design is ready for hardware validation with the PL kernels, C drivers, and V++ scripts also generated automatically. From the user's point of view, AI Engine becomes a customizable hardware accelerator that can be created in seconds without writing a single line of code.

Features

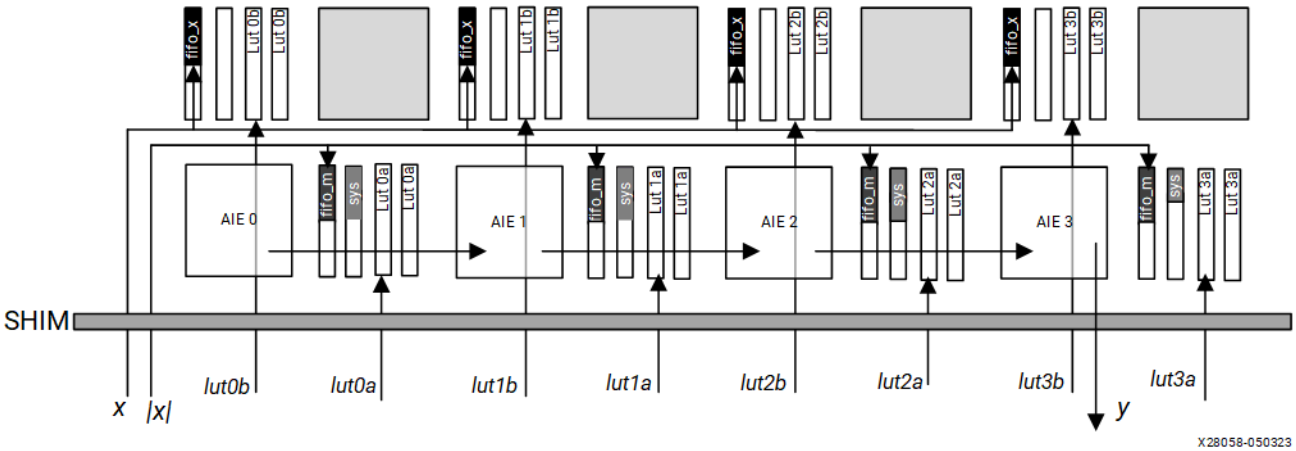
A set of MATLAB® scripts are provided to generate the DPD forward path design on AI Engine with the following features:

- Up to four phases and 3932.16 MSPS sample rate
- Up to 256 LUTs of 256 entries each
- Glitch-free DPD LUT update and switch in runtime
- Locations of non-linear terms can be specified by the user at compile time
- PL kernels, C drivers, and V++ scripts are provided for hardware validation
- User-friendly FIFO-like data interfaces for easy integration with user logic
- Generic Makefile and Tcl scripts reusable by new designs with minor modification.

Monophase DPD on AI Engine

For AI Engine running at 1 GHz, monophase DPD supports up to 983.04 MSPS sample rate. Every AI Engine can process up to 16 LUTs, and the example shown in the following figure can have up to 64 LUTs. The computation results of one AI Engine are in 48-bit I and 48-bit Q format transferred to the next AI Engine via the cascading bus until the end of the chain where the final accumulation result is shifted, rounded, and saturated to 16-bit I and 16-bit Q format. The input data and the magnitude are broadcasted to all AI Engines but might be consumed at different times depending on the delays of the LUTs. DMA FIFOs are inserted in the data buses to absorb the differences in delays and avoid deadlocks. The magnitude values are used to address the data memory. Because AI Engine can perform two memory loads of 256 bits in one clock cycle, two double memory banks are allocated for up to (256 × 2/32 = 16) LUTs per AI Engine.

Figure 2: **Monophase DPD Implementation on AI Engine**



A single AI Engine can compute the following mathematical equation at 983.04 MSPS throughput at 1 GHz clock.

Equation 2: **GMP Terms Mapped to One AI Engine**

$$y_n = \sum_{k=0:3} \left(\underbrace{x_{n-(d+h_k)} \cdot \mathrm{LUTA}(k, |x_{n-(m+k)}|)}_{\text{Upper Left}} + \underbrace{x_{n-(d+h_k+r)} \cdot \mathrm{LUTB}(k, |x_{n-(m+k+s)}|)}_{\text{Upper Right}} \right.$$
$$\left. + \underbrace{x_{n-(d+h_k)} \cdot \mathrm{LUTC}(k, |x_{n-(m+k+v)}|)}_{\text{Lower Left}} + \underbrace{x_{n-(d+h_k+r)} \cdot \mathrm{LUTD}(k, |x_{n-(m+k+v+s)}|)}_{\text{Lower Right}} \right)$$

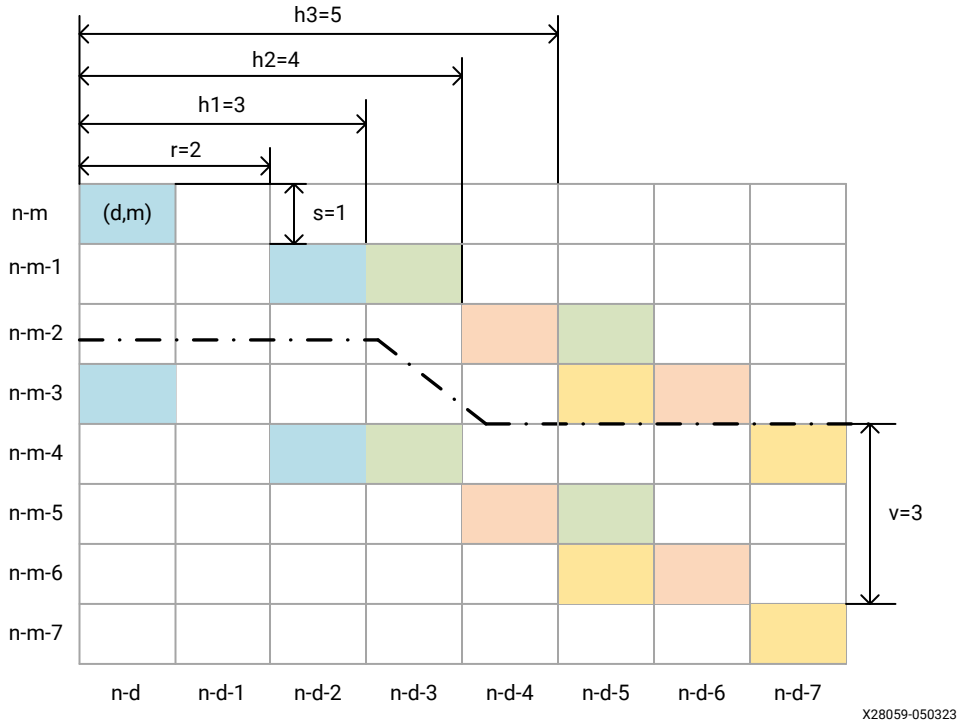
The LUT configuration is determined by the parameters {d, m, v, r, h0, h1, h2, h3}. Without loss of generality, h0 is always selected to be 0 and the set of parameters {d, m, v, r, h1, h2, h3} are visualized in the following figure.

- 16 LUTs must consist of four identical parallelograms which are shown in different colors in the figure.
- The leftmost LUT has a sample delay d, and that of the upper left LUTs of the other three parallelograms are d + h1, d + h2, and d + h3, respectively.
- The top LUT has a magnitude timing offset m, and that of the upper left LUTs of the other three parallelograms are m + 1, m + 2, and m + 3, respectively.
- The lower LUTs of each parallelogram have an extra magnitude delay of v.
- The LUTs on the right-hand side have an extra magnitude delay of s and sample delay of r.

The above parameters determine the LUT configuration without ambiguity. Moreover, these parameters must satisfy the following requirements for a successful mapping to AI Engine:

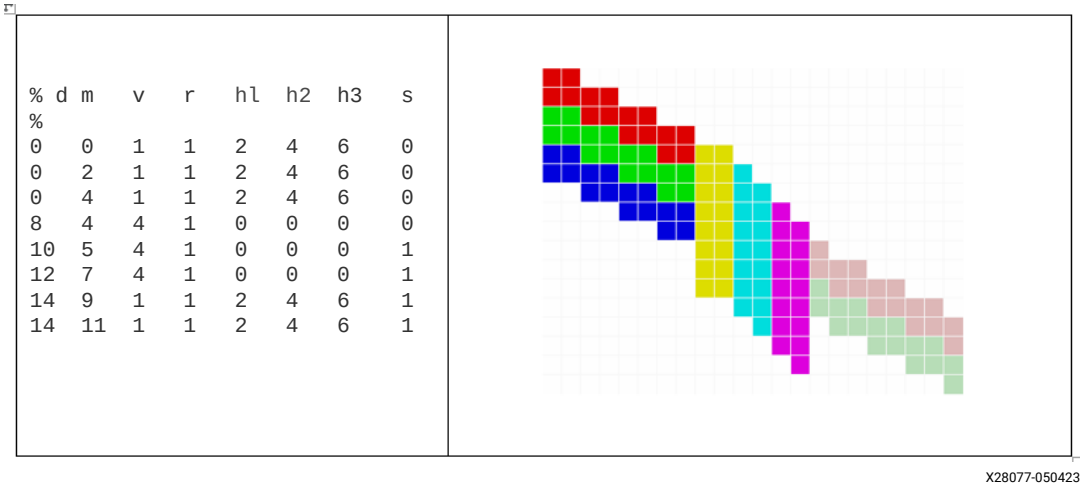
- s = 0 or 1
- v ≤ 6
- Define H'= {0, 1 – h1, 2 – h2, 3 – h3}. max(H') – min(H') + r ≤ 5

Figure 3: **Diagram of Parameters for a Single AI Engine LUT Configuration**



It has been checked that a total of 2400 legitimate LUT configurations can be described by the parameters {d, m, v, r, h1, h2, h3} and mapped to one AI Engine. As an example, the following lines describe a LUT configuration shown in the following figure, where LUTs in each color are mapped to the same AI Engine.

Figure 4: **Example of Parameters for LUT Configuration**



Polyphase DPD on AI Engine

When the sample rate exceeds 983.04 MSPS, a polyphase DPD can be constructed with the monophase design as building blocks. More specifically, Equation 1: DPD GMP Model can be rewritten into the following format where P is the number of phases.

Equation 3: **DPD GMP Model for Polyphase Architecture**

$$y_n = \sum_{d,m} x_{n-d} \cdot \mathrm{LUT}_{d,m}(|x_{n-m}|)$$

$$= \sum_{q=0}^{P-1} \sum_{s=0}^{P-1} \left(\underbrace{\sum_{d_q} \sum_{m_s} x_{n-q-P\cdot d_q} \cdot \mathrm{LUT}_{q+P\cdot d_q, s+P\cdot m_s}(|x_{n-s-P\cdot m_s}|)}_{\text{Terms of One Phase}} \right)$$

where P is the number of phases.

The following figure visualizes the equation by coloring possible LUT locations of the phases differently. For the case of P = 3, there are nine possible combinations of three data phases and three magnitude phases, so at least nine AI Engines are required for each phase of output. There are three output phases, and a total of 9 × 3 = 27 AI Engines are required. Because each AI Engine processes 16 LUTs, the tri-phase DPD can support at least 16 × 9 = 144 LUTs. The following figure shows one possible implementation on AI Engine. The floorplan looks like three monophase DPDs stacked on top of each other with slightly different routing for data and magnitude.

Figure 5: **Diagram of Polyphase DPD (P = 3)**

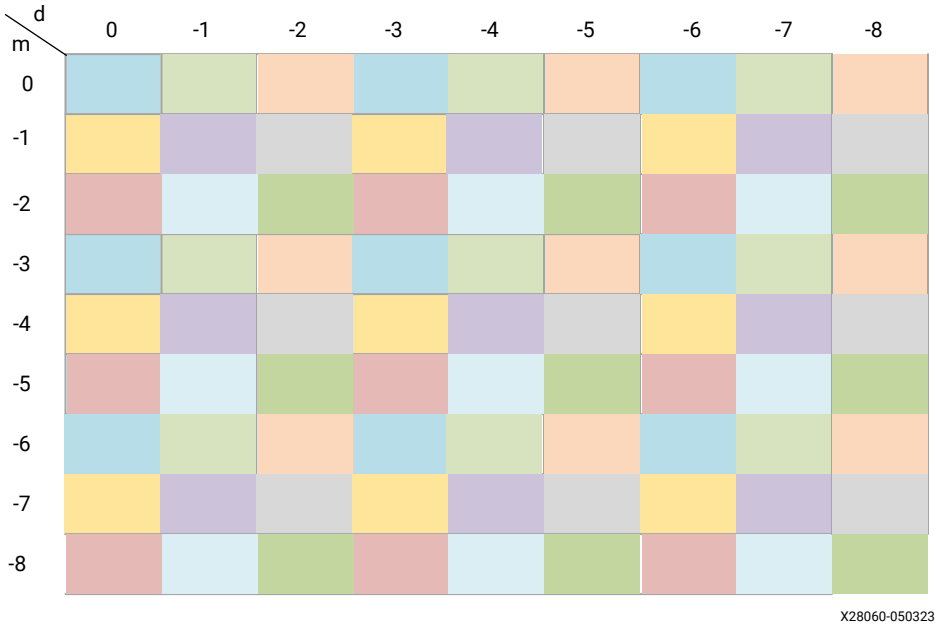
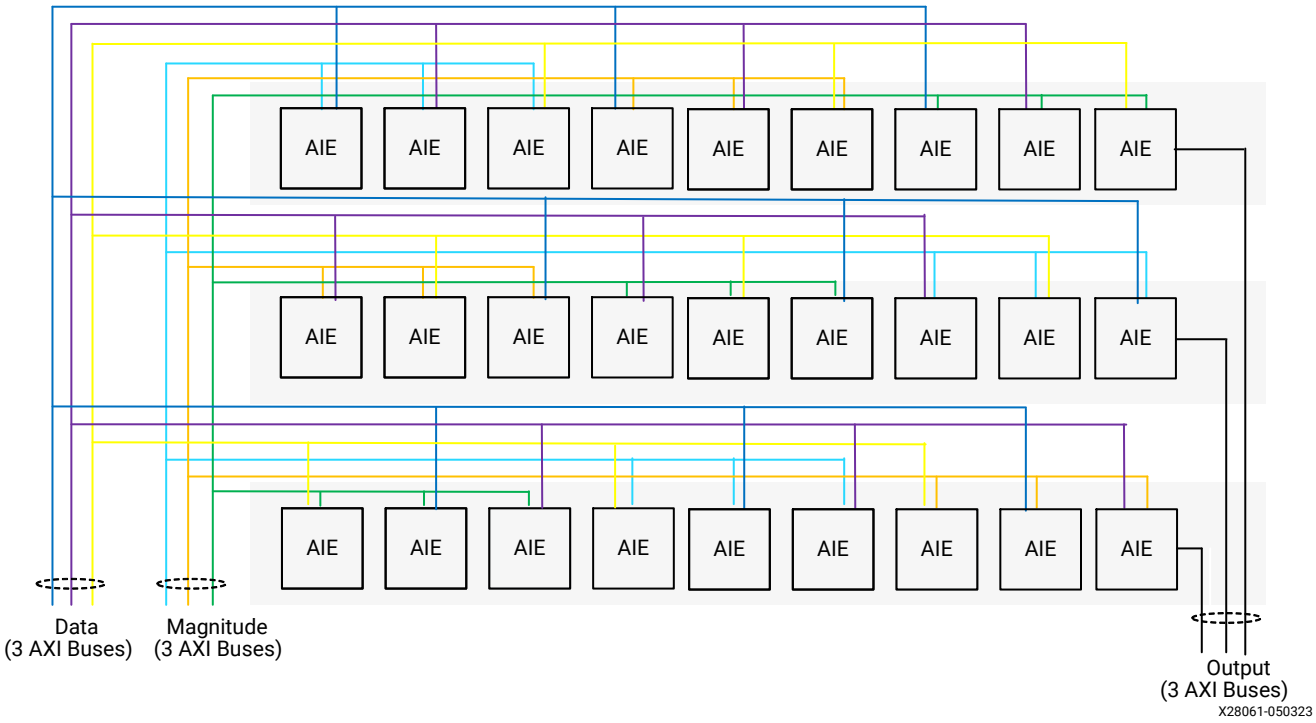


Figure 6: **Diagram of Tri-phase DPD**



For the polyphase DPD, in the LUT configuration file, two more columns are added in each line to specify the phase of data and magnitude. As an example, the following figure shows a case of quad-phase DPD where (a) is the configuration file, (b) is the LUT configuration plotted by MATLAB, and (c) is the AI Engine compilation result using the code automatically generated by MATLAB.

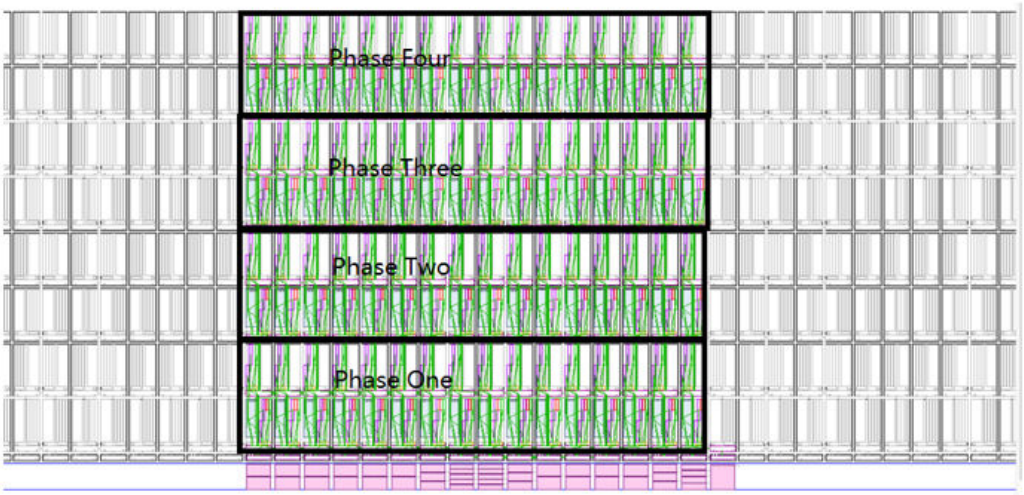
Figure 7: **Diagram of Quad-phase DPD**

(a)

```
% d m   v   r   hl  h2  h3   s  Phd Phm
%
0   0   1   1   2   4   6    1    0 0
0   0   1   1   2   4   6    1    1 0
0   0   1   1   2   4   6    1    2 0
0   0   1   1   2   4   6    1    3 0
0   0   1   1   2   4   6    0    0 1
0   0   1   1   2   4   6    0    1 1
0   0   1   1   2   4   6    0    2 1
0   0   1   1   2   4   6    0    3 1
0   0   1   1   2   4   6    0    0 2
0   0   1   1   2   4   6    0    1 2
0   0   1   1   2   4   6    0    2 2
0   0   1   1   2   4   6    0    3 2
0   0   1   1   2   4   6    0    0 3
0   0   1   1   2   4   6    0    1 3
0   0   1   1   2   4   6    1    2 3
0   0   1   1   2   4   6    1    3 3
```

(b)

(c)



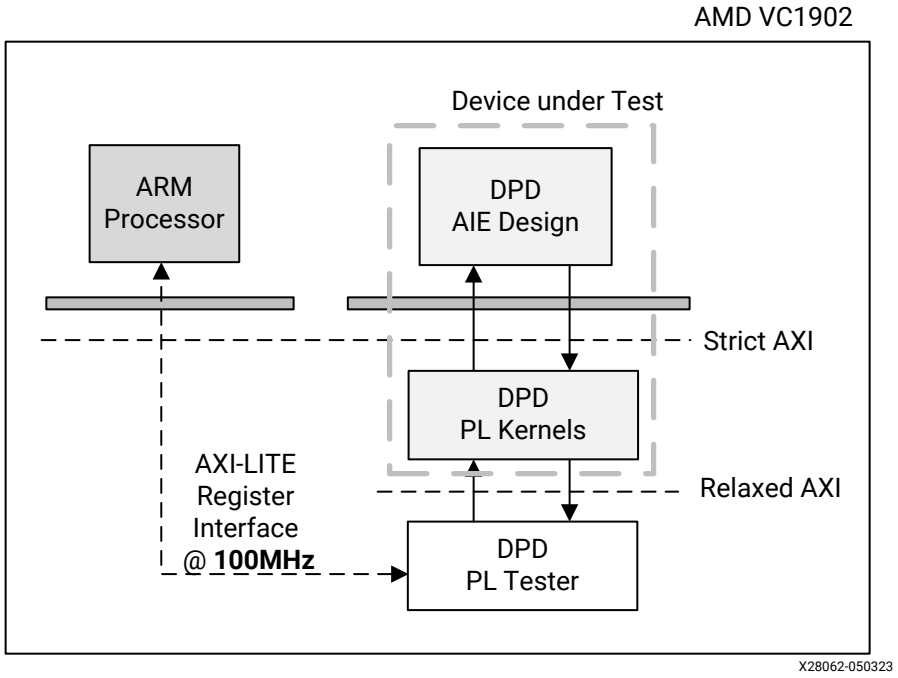
X28076-050423

Hardware Validation

In the DPD design, the throughput of each AXI bus is 983.04 MSPS, which is 98.3% of the theoretical limit of 1 GSPS. The accuracy of AI Engine simulation is around 5%, which is good enough for the designs but not for DPD which can tolerate no more than 1.7% throughput degradation. A hardware validation design is automatically generated by the MATLAB scripts to enable quick validation on an AMD VCK190 board.

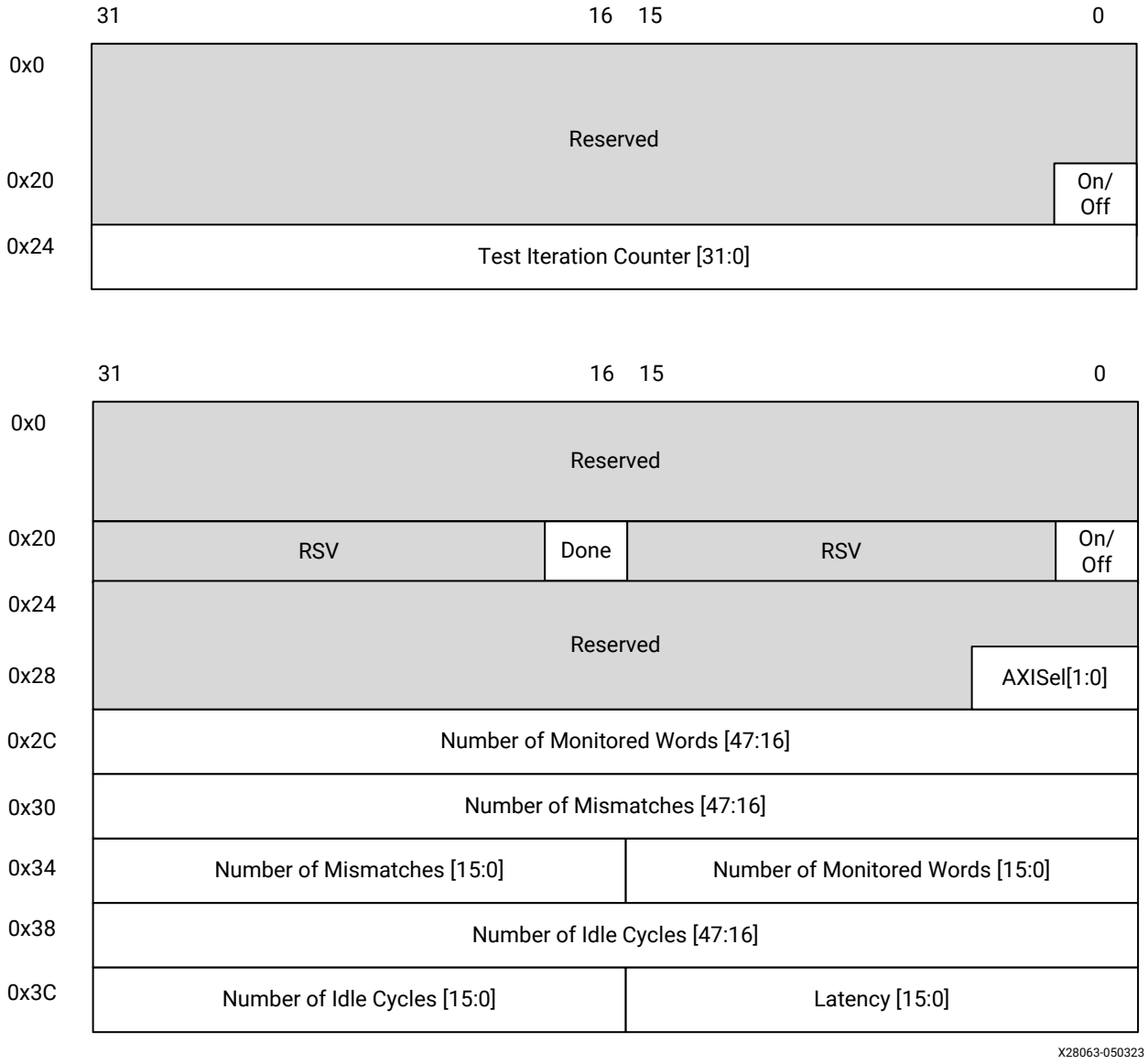
The following figure shows a diagram of the DPD validation environment similar to that in *Arbitrary Resampling Filter Design* (XAPP1373). The AI Engine and PL portions of the DPD design are packaged as kernels, as is the tester which drives the input ports of the device under test (DUT) using a pre-stored stimulus and monitors the output AXI bus with the reference test vector. Throughput and latency are measured by the PL tester and recorded in a set of registers accessible by the processor via the AXI4-Lite interface. At the end of the test, the results are summarized and output through a COM port.

Figure 8: **DPD Design Validation Environment**



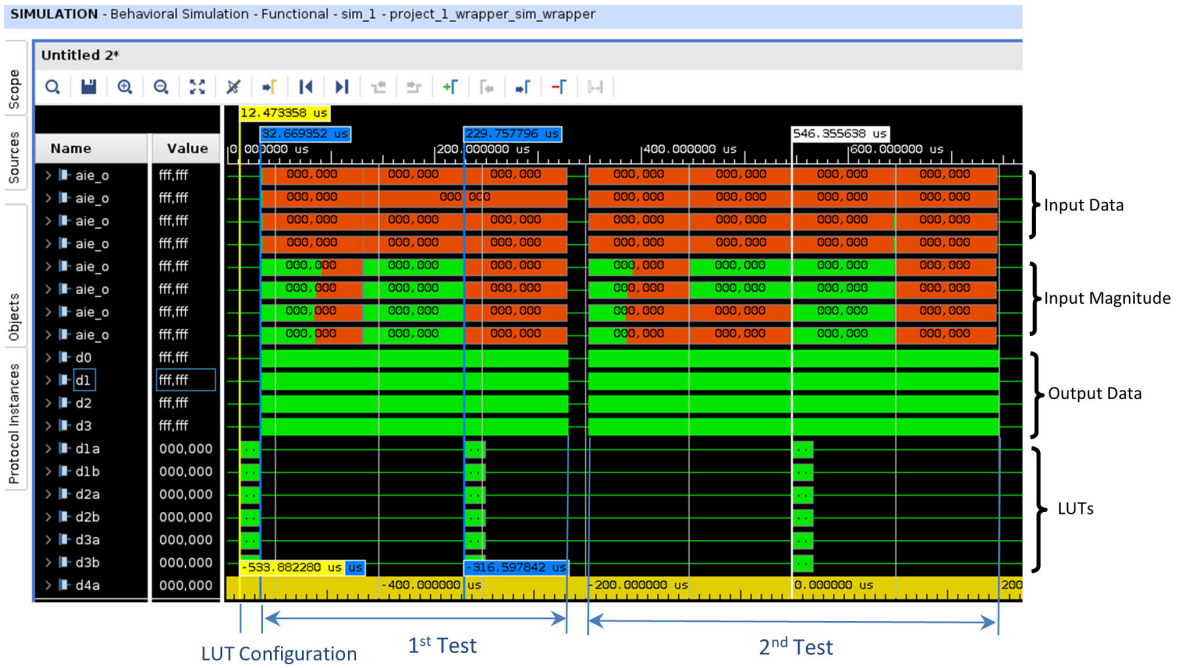
The DPD tester consists of two kernels, `tst_din` and `tst_dout`. The `tst_din` kernel controls the test process via the on/off register and reports the number of data being sent via the `TestIterationCounter` register. One iteration is 122880 samples for each AXI bus. The `tst_dout` kernel collects the outputs and compares them with the golden reference generated by MATLAB. All mismatches are recorded by an error counter. Every output AXI bus has one such monitor which shares the same register map and can be selected by the field `AXISel[1:0]`.

Figure 9: **DPD Tester Kernel Register Map**



Debugging with waveform views in a software simulation environment is much easier than doing so directly on hardware with limited visibility. As noted in *Versal Adaptive SoC System and Solution Planning Methodology Guide* (UG1504), V++ supports PS + PL + AI Engine co-simulation and uses AMD Vivado™ XSIM as the GUI to display waveforms on which latencies of various signals can be measured. The PL kernels on the data path run under a 250 MHz clock with 128-bit data buses, and those on a control path like LUT configuration and register maps are under a 100 MHz clock with a 32-bit data bus. The following figure shows that it takes 20.4 µs for all the LUTs to be initialized before DPD starts to accept input data. The output is available in 0.6 µs for the first test and 0.8 µs for the second test onward. During each test, LUT switching is performed in runtime to make sure no data discontinuity results from the switching.

Figure 10: **PL + PS + AI Engine Co-simulation Waveform for Quad-phase DPD at 3932.16 MSPS**



After the design passes software verification, more comprehensive and longer tests are performed on the VCK190 evaluation board. By default, VCK190 boards come with VC1902-2MP devices. However, in the test platform, the part number is modified to VC1902-1LP, which is recommended for customers who prioritize power efficiency. The software running on the Arm® processor starts and stops the test ten times, from 123456 iterations (12 billion input samples) in the first test with an increment of 98765 iterations (9.7 billion samples) in each of the following tests. In the end, a short summary as shown in the following is output via the COM port.

```
-------------------------------------------------------------------------
--                        DPD TEST SUMMARY                             --
-------------------------------------------------------------------------
AXI-S  Latency(us)  Throughput   No.Outputs    Mismatch   Result
-------------------------------------------------------------------------
First Run
-------------------------------------------------------------------------
 0    0.640- 0.640  985.0Msps    3034546176           0   PASS
 1    0.660- 0.660  985.0Msps    3034546176           0   PASS
 2    0.660- 0.660  985.0Msps    3034546176           0   PASS
 3    0.676- 0.676  985.0Msps    3034546176           0   PASS
-------------------------------------------------------------------------
Afterwards
-------------------------------------------------------------------------
 0    0.868- 0.872  985.0Msps  136537128960           0   PASS
 1    0.880- 0.884  985.0Msps  136537128960           0   PASS
 2    0.900- 0.900  985.0Msps  136537128960           0   PASS
 3    0.912- 0.912  985.0Msps  136537128960           0   PASS
-------------------------------------------------------------------------

PASS!
```

The test results confirm all the output samples match the reference test vector stored in ROMs, and the average throughput is higher than 983.04 MSPS on all AXI buses. The latencies measured on the hardware also match the co-simulation results very well. The processing delays of all four AXI buses differ by several clock cycles which can be easily equalized with small FIFOs.

Conclusion

DPD needs to support high sample rates for 5G and 6G wireless communication systems. The LUT coefficient calculation is performed in the processor, magnitude calculation and other control logic can be placed in programmable logic, and the compute-intensive non-linear filters are mapped to AI Engines. AMD Versal™ AI Core devices with AMD Vitis™ software and a set of MATLAB scripts make the design of such complicated heterogeneous systems much easier than before. Users do not have to write a single line of code to get the whole DPD forward-path design generated and tested on hardware. As a result, AI Engine becomes a fully customizable hardware accelerator that can be customized and created in a very short time without any coding.

Reference Design

Download the reference design files for this application note from the AMD website.

Reference Design Matrix

The following checklist indicates the procedures used for the provided reference design.

Table 1: **Reference Design Matrix**

Parameter	Description
General	
Developer name	AMD
Target devices	Versal AI Core devices
Source code provided?	Y
Source code format (if provided)	MATLAB script, AI Engine C code, Verilog, and Makefile
Design uses code or IP from existing reference design, application note, 3rd party or Vivado software? If yes, list.	N
Simulation	
Functional simulation performed	Y
Timing simulation performed?	N
Test bench provided for functional and timing simulation?	N
Test bench format	Verilog and C
Simulator software and version	AI Engine Simulator and XSIM in Vitis 2022.1
SPICE/IBIS simulations	N
Implementation	
Synthesis software tools/versions used	Vivado synthesis
Implementation software tool(s) and version	Vitis 2022.1
Static timing analysis performed?	Y

Table 1: **Reference Design Matrix** *(cont'd)*

Parameter	Description
Hardware Verification	
Hardware verified?	Y
Platform used for verification	VCK190

References

These documents provide supplemental material useful with this guide:

1. *AI Engines and Their Applications* (WP506)
2. *Arbitrary Resampling Filter Design* (XAPP1373)
3. *Versal Adaptive SoC System and Solution Planning Methodology Guide* (UG1504)
4. D. R. Morgan, Z. Ma, J. Kim, M. G. Zierdt, and J. Pastalan, *A generalized memory polynomial model for digital pre-distortion of RF power amplifiers*, IEEE Trans. Signal Process., vol. 54, no. 10, pp. 3852–3860, Oct. 2006.

Revision History

The following table shows the revision history for this document.

Section	Revision Summary
05/24/2023 Version 1.0	
Initial release.	N/A

Please Read: Important Legal Notices

DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

AUTOMOTIVE APPLICATIONS DISCLAIMER

AUTOMOTIVE PRODUCTS (IDENTIFIED AS "XA" IN THE PART NUMBER) ARE NOT WARRANTED FOR USE IN THE DEPLOYMENT OF AIRBAGS OR FOR USE IN APPLICATIONS THAT AFFECT CONTROL OF A VEHICLE ("SAFETY APPLICATION") UNLESS THERE IS A SAFETY CONCEPT OR REDUNDANCY FEATURE CONSISTENT WITH THE ISO 26262 AUTOMOTIVE SAFETY STANDARD ("SAFETY DESIGN"). CUSTOMER SHALL, PRIOR TO USING OR DISTRIBUTING ANY SYSTEMS THAT INCORPORATE PRODUCTS, THOROUGHLY TEST SUCH SYSTEMS FOR SAFETY PURPOSES. USE OF PRODUCTS IN A SAFETY APPLICATION WITHOUT A SAFETY DESIGN IS FULLY AT THE RISK OF CUSTOMER, SUBJECT ONLY TO APPLICABLE LAWS AND REGULATIONS GOVERNING LIMITATIONS ON PRODUCT LIABILITY.

Copyright