RISC-V from toy to tool

extract of my RV slides for the MicroTCA workshop @ DESY 2023

07.12.2023 Jens.Michaelsen@Avnet.eu, Digital FAE at Avnet Silica
(contact me for details on RISC-V, FPGA, Networking/TSN, and other nerdy things ☺)

AVNET SILICA

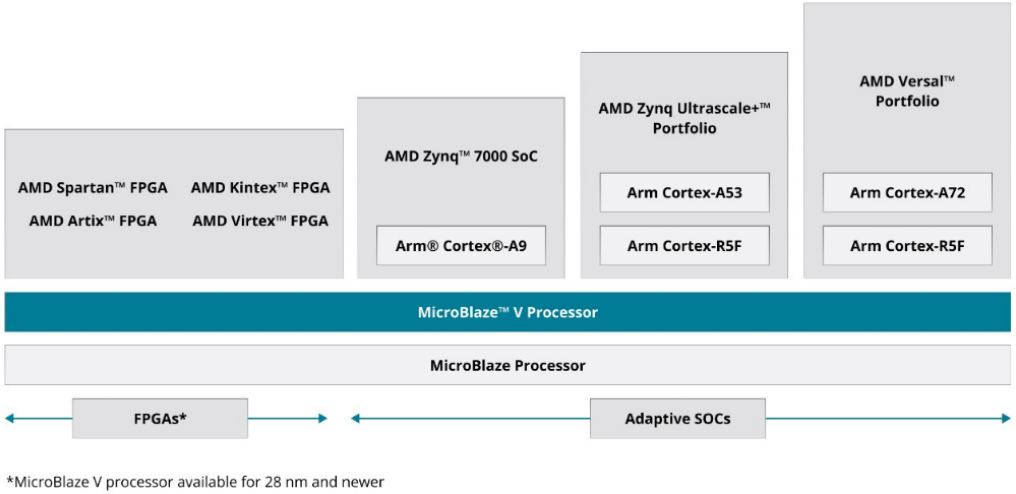
MicroBlaze V

RISC-V on FPGA or “last core standing”

AMD / Xilinx from 2024.1 on supports RISC-V !!

MicroBlaze V Ecosystem :

- RV32 CPU IP Roadmap :
  - MCU, (2024.1)
  - CPU, RTOS (soon after)
  - MPU, Linux (a bit later)
- Design Tools
- Example design
- No plans to stop current MicroBlaze



https://www.xilinx.com/products/design-tools/microblaze-v.html#overview

RISC-V and AMD / Xilinx

MicroBlaze V Features :

- RV32IMAFC
  - Integer Multiply / Divide
  - Atomic Instructions
  - Floating Point
  - Compressed Instructions
  - Bit Manipulation (Zba.b.c.s)
- AXI Streaming
- Interface
  - AXI
  - ACE
  - LMB

Example Design Configurations

 Click to enlarge	**Microcontroller** • 32-bit Processor Core (RV32IMAFC) • JTAG Debug Interface • Tightly Coupled Local Memory • SPI controller • I2C Controller • UART • Interrupt Controller • Timer • GPIO
On the Roadmap	**Real-Time Processor**
On the Roadmap	**Application Processor**

b.t.w. since 2020 there is an unofficial RISC-V debian vivado repo on github
https://github.com/eugene-tarassov/vivado-risc-v !this is not related to AMD!

FPGA and RISC-V Timeline, now „**monopolist**“ a bit like Linux

proprietary cores had their time

Why FPGA vendors move to RISC-V?

- free to use ISA
- Good toolchain, verification, RTOS support
- Linux to speed up SW design
- Simple to implement in FPGA
- No need for own tooling

- One toolchain for
  - FSM
  - MCU
  - MPU
  - DPU
  - Cores in Soft & Hard

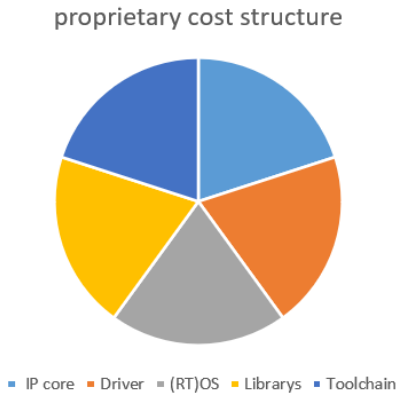
	2012			2016		2020			2024		
Microchip (own+SiFi,MiV)	a	a	a	m5	m5	m5	m5	m5	m5	dark = hard
FPGA L	mi	mi	mi	mi	mi	p5	p5	p5	p5'	light = soft
FPGA E						s5	s5	s5	s5'	
FPGA G							g5	g5	g5	
FPGA I / A	n2	n2	n2	n2	n2	n2	n5	n5	n5	
AMD / Xilinx (own,MicroBlazeV)	µb	µb	µb	µb	µb	µb	µb	µ5	µ5	

Table : who migrated when to RISC-V

RISC-V synergy & savings

Cost model

- IP core
  - Buy or build
- Driver
  - Self, always
- OS/RTOS
  - Self vs. community
- Librarys
  - Self vs. community
- Toolchain
  - Self vs. community



IP synergy

- Go Linux mainstream
  - write drivers for own HW, 10% effort.
  - use full blown Linux stack, 100% payback
- Many RTOS to choose from
- Tools and Librarys
  - Stay up to date
  - Shared effort

RISC-V

Basics

RISC-V ISA

Instruction Set Architecture

What is RISC-V ?

- A free to **use** RISC Instruction Set Architecture
- Standard maintained by RISC-V International

- Intended to be usable for any computing systems
  - RV32I, small MCU footprint
  - …
  - RV64GCV… , MPU number cruncher

1. What is the license model?

The RISC-V ISA is free and open with a permissive license for use by anyone in all types of implementations.

https://riscv.org/faq/

RISC-V Job Sharing

Instruction Set Architecture

RISC-V combines **Scala**r & **V**ector computing

The concept implies „job sharing“

- The **Scalar** unit will take care of the
- flow control job :
  - for(i=0; i<len; i++)

- The scalar unit is mandatory (RVI)
- SPU or MCU/MPU/CPU (many names)
- A RV SPU is 32 Bit or 64 Bit wide

Example:

```
for( i=0; i<len; i++ )
    c[i] = a[i] + b[i];
```

A **Vector** unit can take care of the

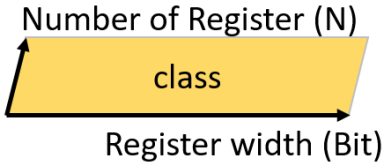
- data flow job :
  - c[i] = a[i] + b[i];

- The vector unit is optional (RVV)
- VPU or DPU/DFP/…
- A VPU is in general >=128 Bit wide
- Latency is not so relevant for VPU,
- this allowes for more pipelining

b.t.w. : in „small“ systems the scalar unit has to do both jobs

RISC-V Geometry

RISC-V Geometry

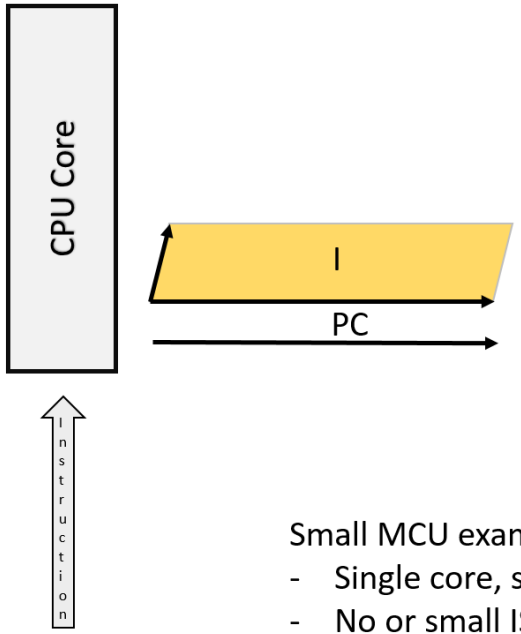


Dimensions of **RISC-V** cores :

- Number of Register
- Register class
- Register width
- Feature support

RISC-V Example 1: Integer Core

Bit => 8 16 32



32 Integer Register

1 Program Counter

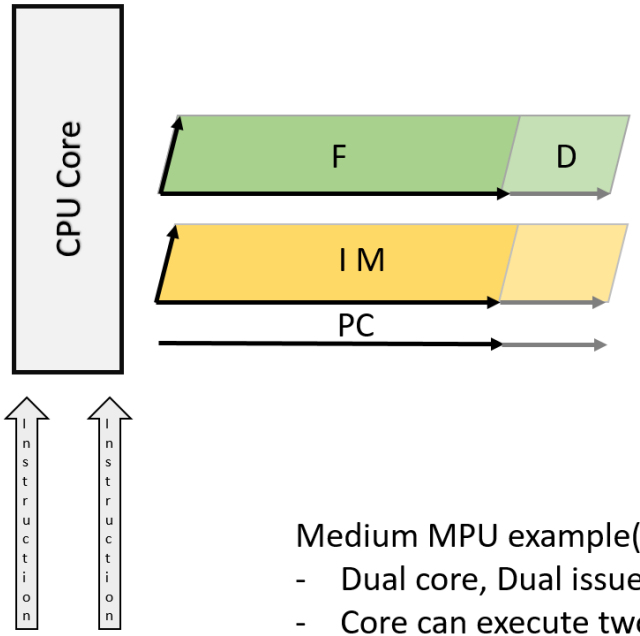
RV32I Controller Class
Features :
- 32 Bit Instruction
- 32 Bit Execution
- 32 Register
- 32 Bit Pointer

Small MCU example(s) :
- Single core, single issue, medium pipelined (1~5) engine
- No or small I$/D$ cache, run from FLASH, SRAM, DRAM
- RTOS support, has no MMU
- Starting with an tiny footprint of only 47 instructions
- „myFirst RISC-V“ student homework/ playground @ GitHub

Integer core options :
- **I** nteger
- **M** ultiply
- **A** tomic
- **B** itmanipulation
- **E** mbedded

RISC-V Example 2: Floating Point Core

Bit => 8 16 32 64



32 Floatingpoint Register FLEN

32 Integer Register XLEN

1 Program Counter XLEN

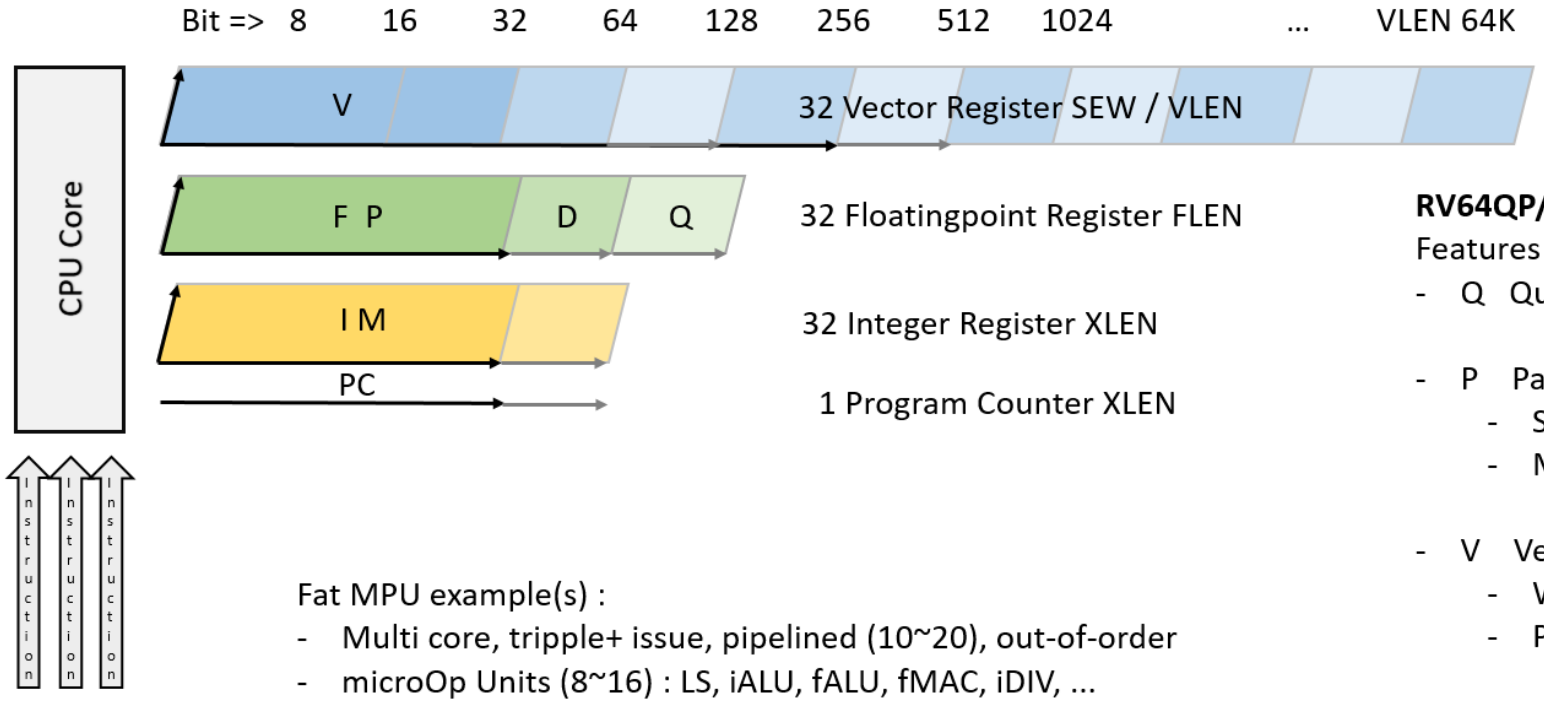
RV64GC Application Class Features :

- G Generic = IMAFD
- I Integer
- M Multiply
- A Atomic
- F Float 32 Bit
- D Double 64 Bit
- C Compressed
- S Supervisor +MMU

Medium MPU example(s) :

- Dual core, Dual issue, pipelined engine (5~15)
- Core can execute two instructions per clock
- I$/D$ cache and DRAM/RAM
- Supervisor & MMU support for Linux
- Execution length : XLEN = FLEN = 64 Bit

RISC-V Example 3: Vector Processor Core



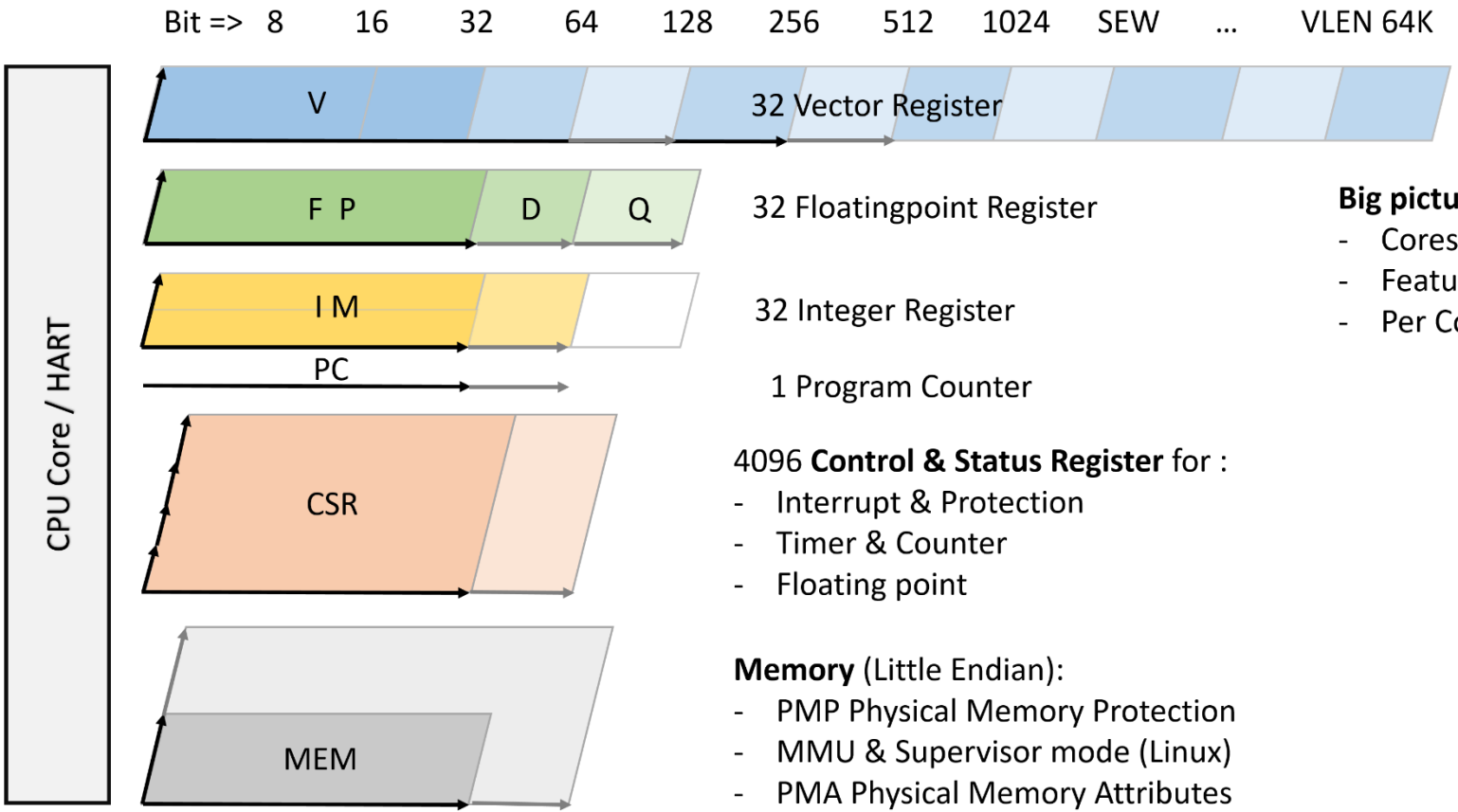
RV64QP/V Compute Class Features :

- Q Quad Float 128 Bit
- P Packed SIMD
  - Single Instruction
  - Mutiple Data
- V Vector Engine
  - Wide Data
  - Processing

Fat MPU example(s) :

- Multi core, tripple+ issue, pipelined (10~20), out-of-order
- microOp Units (8~16) : LS, iALU, fALU, fMAC, iDIV, ...
- Vector processing engine
- 32 visible Vector register, each 256~512 bit wide
- P+V still in status draft/review, as of early 2022

RISC-V Cores total view



Big picture :
- Cores view
- Feature dependent
- Per Core/Thread

Random Details on other Products

Random Product Details

Gemini II

- Compute-In-Memory CIM
- 5.8 PETA OPs @ 1bit
- 184 TERA OPs @ int8
- ~102 MByte

Versal Premium

- Adaptive Compute Acc. Platf.
- 1.0 PETA OPs @ int4
- 270 TERA OPs @ int8
- ~140 Mbyte (PL)

Quality Timing

- < 0.1 DPPM Quality
- 1960 (Mio hours) MTBF

Analog Conversion

- ADC 1x 10 bit @ 12.8 GSps
- DAC 2x 12 bit @ 12 GSps

RF-SiP

- ADC + DAC 4x @ 64 GSps
- + Versal Core VC1902
- + 4GB LPDDR4

GSI **Gemini I+II CIM** devices

CIM compute-in-memory:

- Reduce power
- System size
- System cost
- Operational cost

Processor Comparison

	Gemini-I APU	Xeon 8280	Nvidia A100	Graphcore
Compute Cores	2 million x 1 bit	28 x 2 x 512 bits	104 x 4096 bits	1216 x 64 bits
Compute Speed	600 MHz	2.7 GHz	1.4 GHz	1.6 GHz
Peak Compute*	38 TOPS	10 TOPS	75 TOPS	16 TOPS
On-Chip Memory	12MB L1	38.5MB L3	40MB L2	300MB L1
Mem Bandwidth	39TB/s	1TB/s	7TB/s	16TB/s
Power	80W TDP	205W TDP	400W TDP	150W TDP

*Trillions of 8-bit ADD operations per second (TOPS)

Gemini II just launched :

- 184 TERA OPS (8bit) or 5.8 PETA OPS (1bit) @ 1.4 GHz

Avoid exta data transfer and save ~90% of total power.

Shift from central-processing to distributed computing.

Break down the performance limiting memory wall.

Contact : jens.michaelsen@avnet.eu

SiP 1) System.in.Package with ADC & DAC

SiP w/ digital & analog:

- 2 x Cortex A72
- ~2 k DSP blocks
- 400 Embedded AI engines
- 4GB LPDDR4
- 4 x ADC receive channels @ 64 GSPS
- 4 x DAC transmit channels @ 64 GSPS

Save up to 75 % area of your next design.

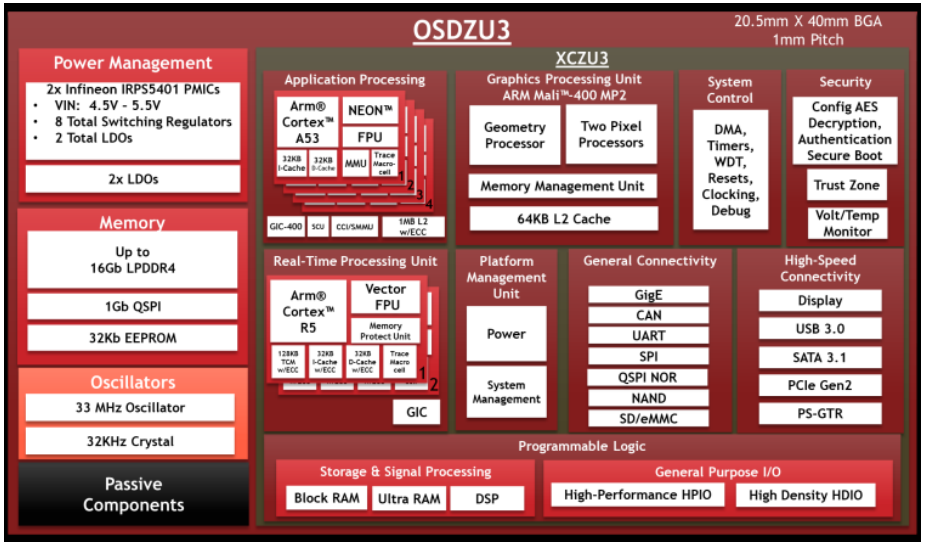
Interested? Get in touch!

Contact : jens.michaelsen@avnet.eu

SiP 2) System.in.Package with Zynq UltraScale+

SiP ZU3EG based:

- 4 x Cortex A53
- 2 x Cortex R5F
- 2GB LPDDR4
- PCIe® Gen2 x4
- 2x USB 3.0
- 4x 1Gig Ethernet
- …



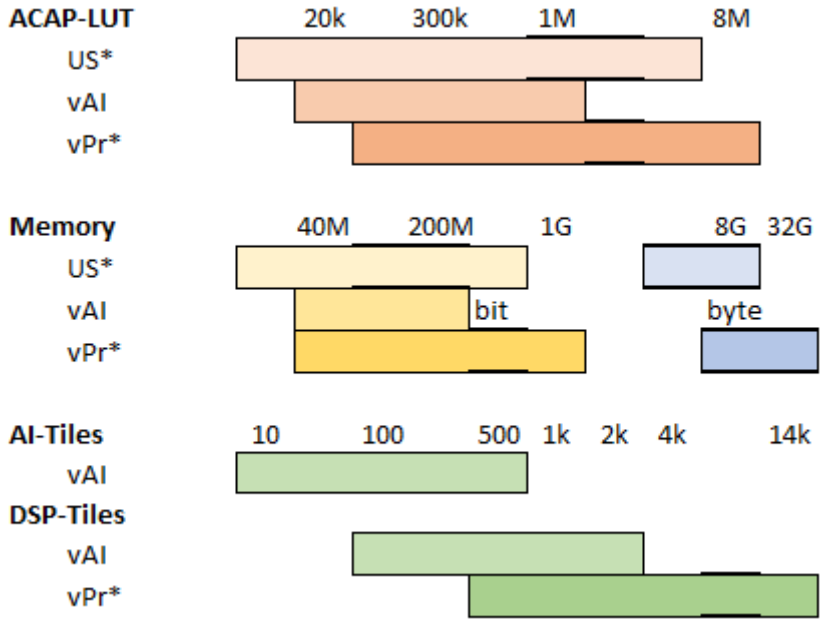
Save up to 57 % area of your next design.

Interested? Get in touch! Contact : jens.michaelsen@avnet.eu

Versal & UltraScale+ range of some features

Families:

- US*
  - UltraScale+
  - MPSoC & **HBM**
- vAI
  - Versal AI Edge
  - Versal AI Core
- vPR*
  - Versal Prime
  - Versal Premium
  - Versal **HBM**



- B.t.w. this is my personal data collection, may be unpricise, or even be missing something
- Intended to pre select the product family to further investigate, use at your own risc

Thank you!