arm

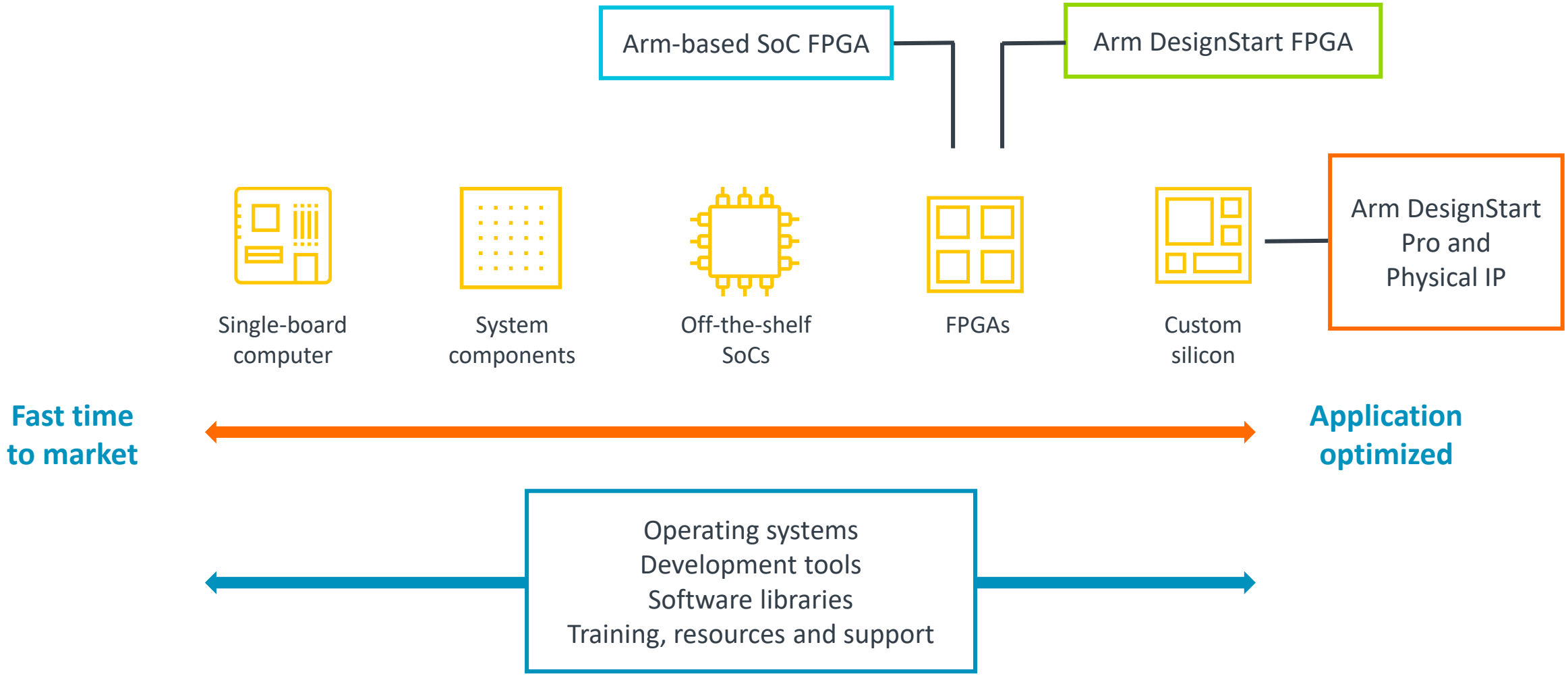

XILINX

# Implementing Arm Cortex-M1 & Cortex-M3 in Xilinx FPGAs

Chris Shore, Director Embedded Solutions

ESC Boston 2019

# Consistent architecture lowers development costs






arm

# Speed-up design with fast access to Arm IP and ecosystem

## DesignStart for custom SoCs/ASICs

- ✓ Cortex®-M0 and Cortex-M3 CPUs plus subsystem for no upfront fee
- ✓ Cortex-A5 CPU + system IP: lowest-cost route to Linux-capable design

## DesignStart FPGA

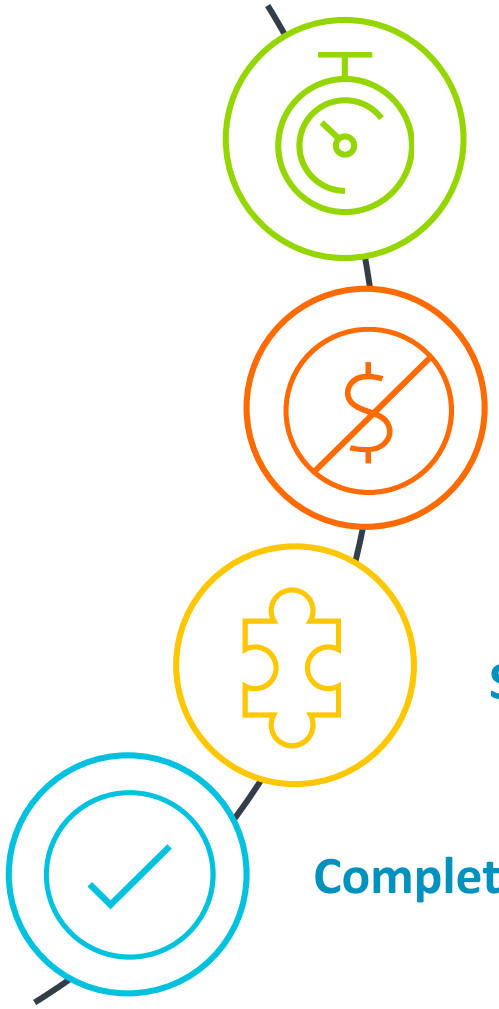
- ✓ Cortex-M1 and Cortex-M3 CPUs
- ✓ Soft IP in FPGA, integrated with partner design flow
- ✓ No license fee, no royalties



arm

# Accessing Cortex-M1 and Cortex-M3 on Xilinx FPGA

arm
DESIGNSTART FPGA



**Fast:** easy and instant access to Cortex-M1 and Cortex-M3 CPUs

**No cost:** no license fee, no royalty

**Simple to use:** integrated with partner tool flows

**Complete:** accelerated software design with Arm Keil MDK

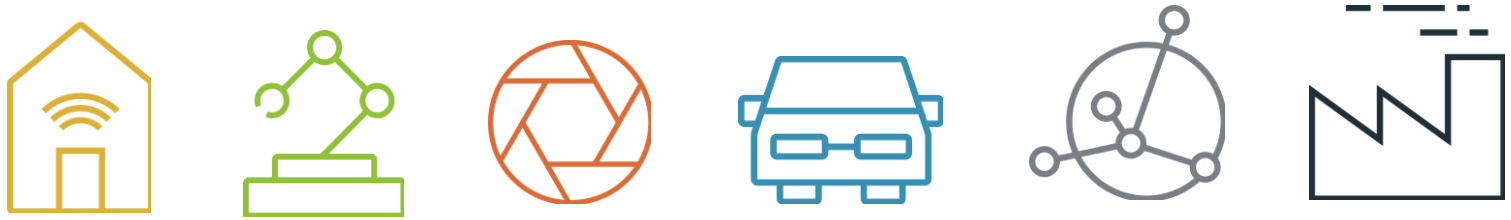
# Choice of Cortex-M Processors for FPGA

## Cortex®-M1

- FPGA-optimized version of Cortex-M0
- 32-bit processing in the smallest area
- For control logic, management and area constrained applications

## Cortex-M3

- General purpose 32-bit processor
- Balanced performance and area
- For diverse embedded and IoT applications



Exceptional code density

Simplified software development & vendor-independent CMSIS abstraction layer

Supported by the broadest technology ecosystem of software, tools & services

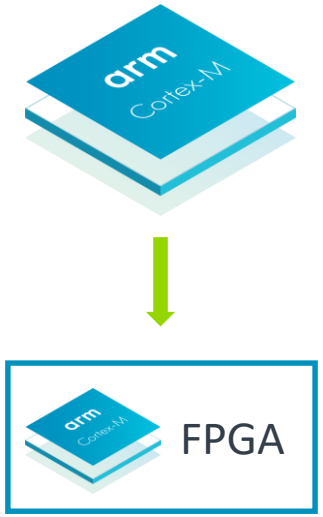



arm

# Rapid time to market with simplified development flow

## Design hardware

Simple drag-and-drop integration of CPU




## Develop software

Benefit from broadest embedded ecosystem

Reuse existing code

Access widest range of third party software

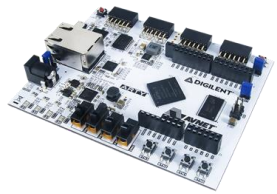
arm KEIL

IAR SYSTEMS

## Deploy on FPGA

Deploy to any development board

Pre-integrated on development boards





arm

# Developer Base, Tools and Development Resources

## Thriving developer base

**350k+** Mbed OS registered developers

**2+ million** Eclipse/GCC (Arm) downloads in 1 year

**8.5+ million** CMSIS pack downloads in 1 year

## Largest choice of proven OS and tools


**40+** RTOS

**20+** IDE compiler

**21+** Debug & trace

## Largest open-access development resources

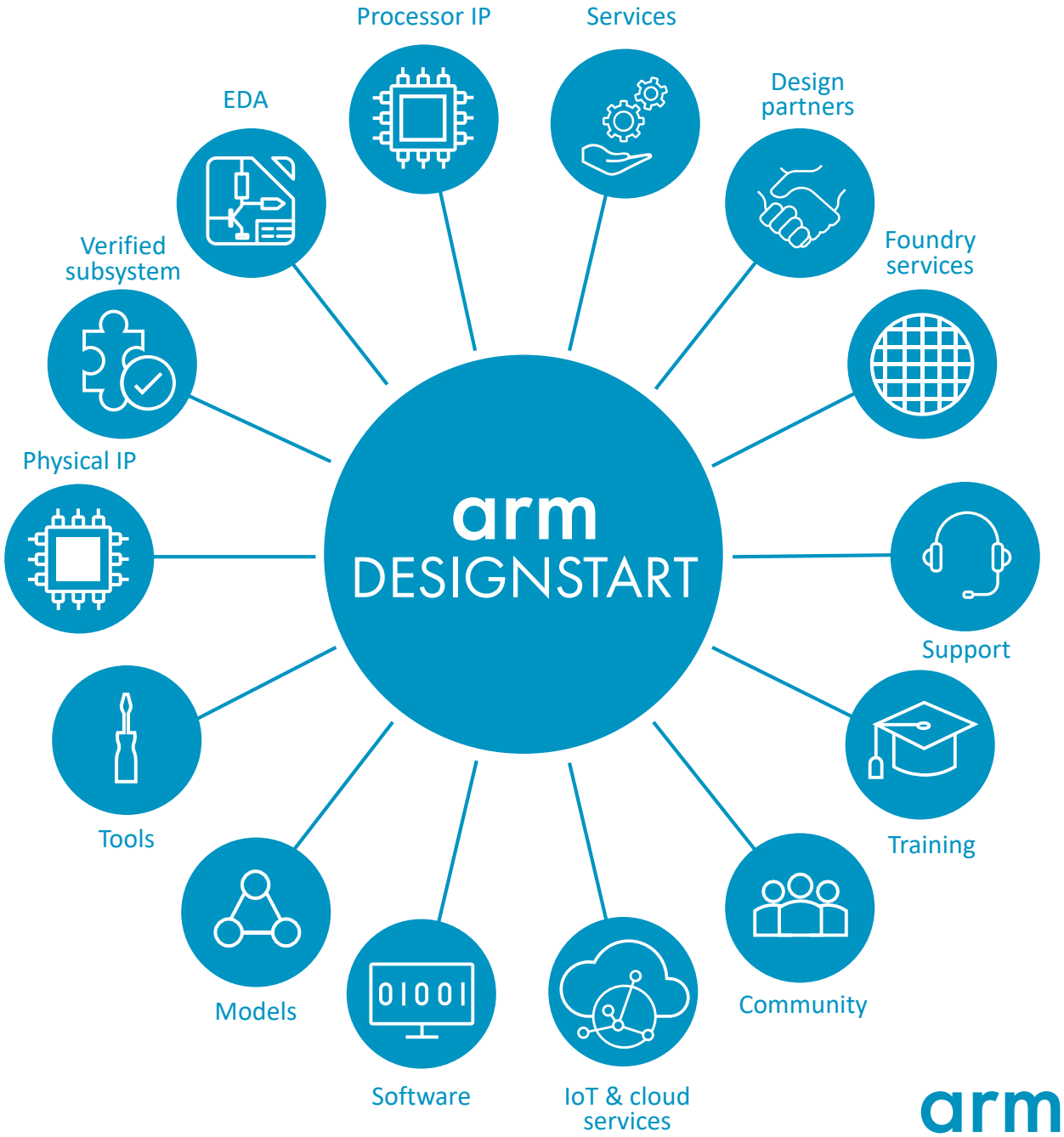
arm Community
arm Developer

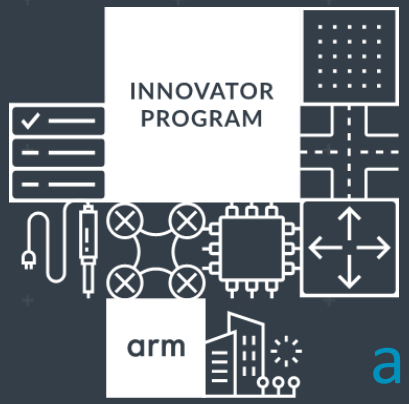
*1000's of how-to guides, articles, and online development resources*



arm

# Much more than a CPU for silicon success






arm.com/innovation

Technical experts who evangelize Arm's story in AI, security, drones, robotics, IoT and more.

# Why an Embedded Solution from Xilinx?

> Need to **accelerate software functions in hardware** to offload the processor

> Need a **companion FPGA** with my ASSP/MCU

> Need **product differentiation** through custom engines or analytics

> Need to extend **I/Os** and/or **protocol support**

> Want a **future-proof** system against evolving standards

If any of these applies to you, Xilinx has embedded solutions for you





XILINX.

# MicroBlaze vs. Arm Cortex-M

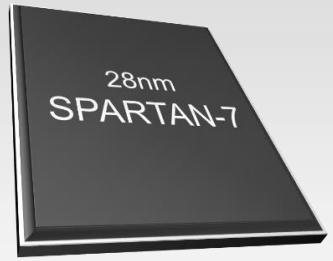
- **Why Arm?**
  - Existing Arm Experience
  - Minimal porting Arm code to MicroBlaze™
    - If coming from Arm® Cortex®-M and use Keil MDK, then only minimal code porting required
  - Exception code density
    - May require less BRAMs
  - Ecosystem
- **Why MicroBlaze?**
  - Performance
    - MB can run 2-3x faster than Cortex-M
    - FPU support, emulating M4 configuration
  - Area
    - MB is 10% less area for MicroController version
    - MB is 66% less area for Real-Time Version
  - Better SDK Integration and Debug Support

# Cost-Optimized Portfolio in Full Production

SPARTAN

I/O Optimized

Spartan-7




2.5X Performance/Watt Advantage

ARTIX

Transceiver Optimized

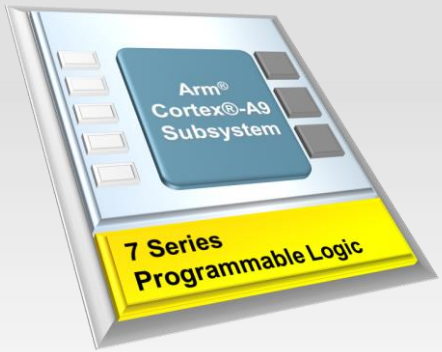
Artix®-7


Lowest Cost Transceiver-Based Device with Two New Devices

ZYNQ

System Optimized

Zynq®-7000




3X Growth Over Two Years

arm Cortex-A9

arm Cortex-M



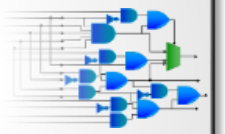


XILINX

# Spartan®-7 FPGA Overview

## 2.5X Perf/Watt

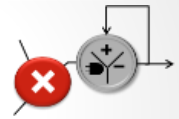
50% lower power and 30% faster than previous generation



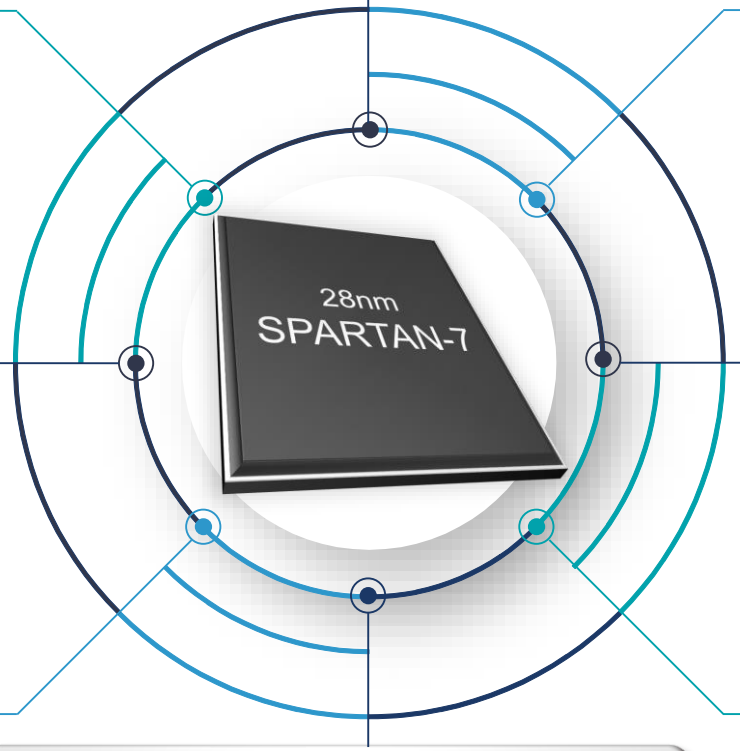
## Internal Memory


- Up to 4.2Mb total
- Perfect for code storage & clock domain crossing, FIFOs

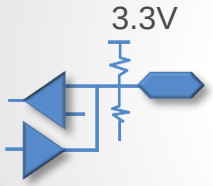
## DSP



- Wide 25x18 MACs
- Up to176GMACs for DSP and Image Filters especially video motion estimation

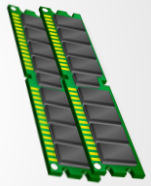



## Flexible I/O




- Any-to-Any Connectivity
- Up to 1.25G LVDS
- Up to 400 I/O

## External Memory



- Support for DDR3-800, DDR2, LPDDR2 up to 1GB densities

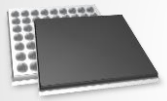
## SECURITY

- Encryption & Authentication
- AES256 CBC & SHA-256

## ADC & System Monitor

- Dual 12-bit 1MSPS ADC with 17 input Analog Mux
- Thermal monitoring

## Small Form Factor



Only 28nm device in an 8x8mm package with 100 I/O



XILINX.

# System-Wide Safety and Security Across the Portfolio

## > Hardware (FPGA Fabric)

- >> AES-256 encryption[1] → anti-cloning & reverse eng.
- >> SHA-256 authentication → Ensures trusted source
- >> Temp/Volt. monitor → Flags 'out-of-spec' condition
- >> Isolated Design Flow (IDF) → Fault containment[2]

## > Software (Zynq® SoC)

- >> Secure boot, protects from attack at startup
- >> Arm® TrustZone[3] to isolate 'main' OS from secure OS
- >> Memory protection against malware injection
- >> Rich ecosystem of run-time Security IP


1: NIST-Approved (National Institute of Standards and Technology)
2: Physical separation of safety-critical regions of the design
3: A compromised OS cannot access 'secured' data in the secure OS



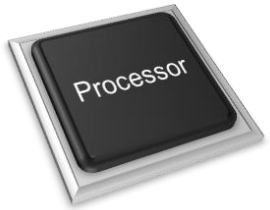


XILINX.

# Devices Designed To Minimize System Cost

Replace Microcontroller, Processor, & External Cache Memory

**($1 to ~$15 Savings)**



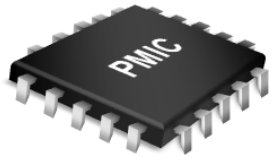

Replace External DSPs

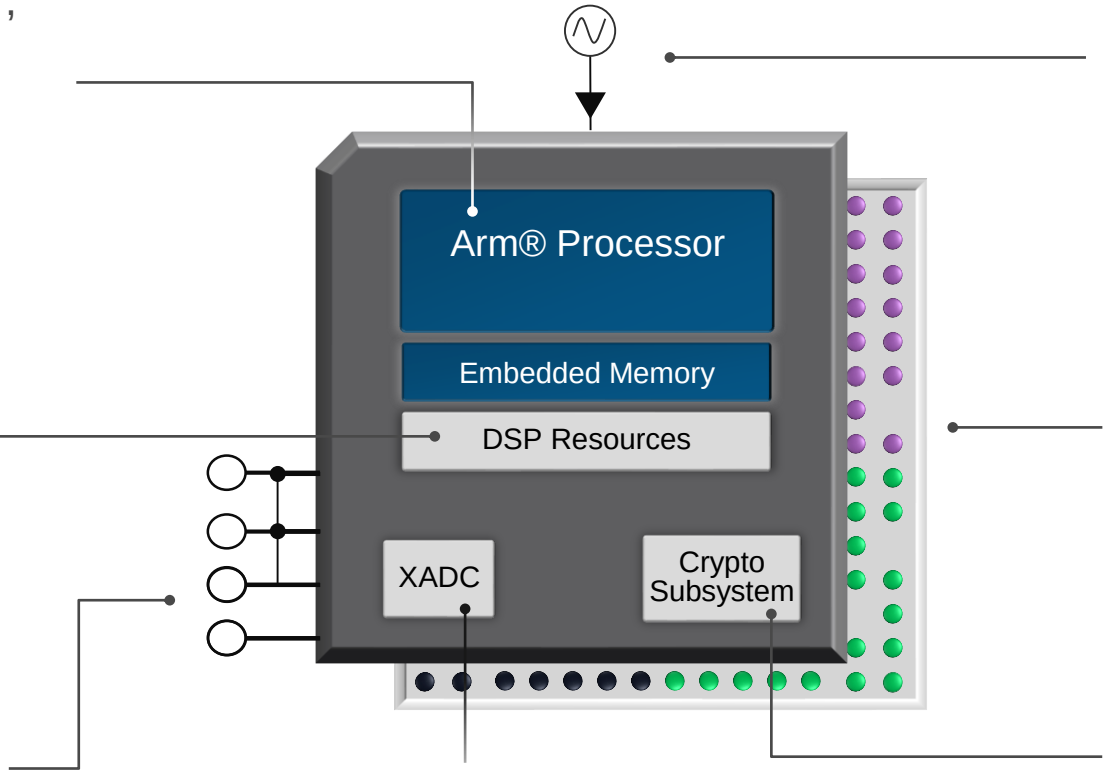
**($1 - $2 Savings)**


PMIC Support & Flexible Rails Reduce # of Power Supplies

**($1 - $2 Savings)**

*per Supply*




Integrated AMS Eliminates External ADCs

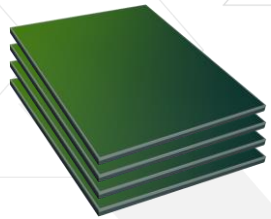
**($1 to $2 Savings)**

Clocking Structured Reduces # of On-Board Oscillators

**($1 - $2 Savings)**


Packaging & Pinout Reduces Board Layers

**(Minimize PCB Cost)**



Encryption & Authentication Eliminates Security Device

**($1 to $2 Savings)**






XILINX.

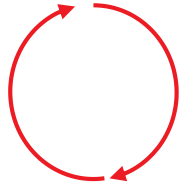
# Time to Market Advantage with Arm and Xilinx

## µP + FPGA

*Months*

Choose µP → Choose FPGA → Partition Design → Determine Interface & Bandwidth → Board Design & Layout → Debug Board → System Level Programming → First Ship

- ✕ Must deal with IC Sequencing
- ✕ SI/PI issues
- ✕ Maintain Multi-IC Lead Times



XILINX arm

*Hours*

Choose SoC → Start System Programming w/Eval Board

- ✓ No additional device
- ✓ No partitioning
- ✓ No interface design
- ✓ High bandwidth

Board Design and Layout → Debug Board → First Ship

- ✓ System Design in Parallel
- ✓ Fewer ICs, easier debug

**Faster Time-to-Market**

- ✓ Single Vendor,
- ✓ Short Lead Times

# Get Started with Cortex-M on Single Core Zynq-7000S

ZYNQ Arm® Cortex®-A9 +

| Device Name | Z-7007S | Z-7012S | Z-7014S |
|---|---|---|---|
| 7 Series PL Equivalent | Artix®-7 | Artix-7 | Artix-7 |
| Logic Cells | 23K | 55K | 65K |

SPARTAN 7

| Part Number | XC7S6 | XC7S15 | XC7S25 |
|---|---|---|---|
| Logic Cells | 6,000 | 12,800 | 23,360 |

- Most cost-optimized Zynq® device (Z-7007S) features 23K logic cells! (3rd largest Spartan-7)
- Cortex-M1 minimal configuration would consume less than 1/10th of device
- Block RAM can be used as on-chip memory
- Cortex-M Resource Utilization – Example Design

| Default Configuration | Cortex-M3 Arty A7–35T | | Cortex-M1 Arty S7–50T | |
|---|---|---|---|---|
| Freq. Max | 50MHz | | 100MHz | |
| Slice Registers | 4680 | 11% | 2600 | 4% |
| Slice LUTs | 12950 | 62% | 3200 | 10% |
| BlockRAM | 32 | 64% | 24 | 32% |
| DSP | 3 | 3% | 3 | 3% |
| | 64KB – ITCM<br>64KB – DTCM<br>32 IRQ | | 32KB – ITCM<br>32KB – DTCM<br>32 IRQ | |

| Minimal M1 Configuration | Cortex-M1 Arty S7–50T | |
|---|---|---|
| Freq. Max | 100MHz | |
| Slice Registers | 1722 | 3% |
| Slice LUTs | 2400 | 7% |
| BlockRAM | 10 | 13% |
| DSP | 0 | 0% |
| 32KB – ITCM<br>8KB – DTCM<br>8 IRQ | | |





XILINX

# Tools

- **Keil® MDK Tools**
  - Complete support for Arm Cortex™-M, Arm Cortex-R4, Arm7™, and Arm9™ devices
  - MDK-lite: Free, restricted to 32KB
  - Industry-leading Arm C/C++ Compilation Toolchain
  - µVision4 IDE, debugger, and simulation environment
  - Keil RTX deterministic, small footprint real-time operating system
  - TCP/IP Networking Suite USB Device and USB Host stacks
  - Complete GUI Library

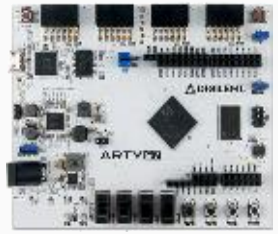

MDK-Arm Microcontroller Development Kit

| | |
|---|---|
| Arm C/C++ Compiler | µVision<br>Project Manager, Editor & Debugger |
| RTX Real-Time Operating System | |
| CAN Interface | File System |
| USB Host | USB Device |
| TCP/IP Networking Suite | GUI Library |

# Getting Started with Cost-Optimized and Zynq UltraScale+ Kits

## ARTY S7
Spartan®-7 S25

**$89**



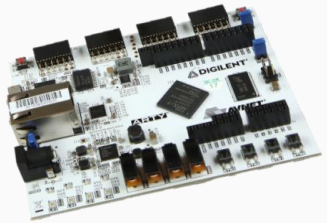
Spartan-7 S50

$109

AVNET
DIGILENT

## ARTY A7
Artix®-7 A35T

**$99**



AVNET
DIGILENT

## ARTY Z7
Zynq®-7000 7Z010

**$149**


Zynq-7000 7Z020

$209

AVNET
DIGILENT

## MiniZed
Zynq-7007S

**$89**

AVNET

## Ultra96
Zynq UltraScale+ ZU3EG

**$249**

AVNET





XILINX.

# Getting Started

# Required Demo Hardware and Software

- **Hardware**
  - Arty S7 Board
    - https://store.digilentinc.com/arty-s7-spartan-7-fpga-board-for-hobbyists-and-makers/
  - Arm® DesignStart DAPLink Board (Optional but recommended)
    - https://developer.arm.com/products/system-design/development-boards/designstart-daplink-board
  - Micro and Mini USB cables
- **Software**
  - Vivado® 2018.2
    - Xilinx Downloads - https://www.xilinx.com/support/download.html
  - Digilent Board files from GitHub
    - Navigate to https://github.com/Xilinx/XilinxBoardStore/tree/2018.2
  - Arm IP from ARM Design Start FPGA
    - https://www.arm.com/resources/designstart/designstart-fpga
  - Arm Keil MDK Tools
    - https://www.keil.com/download/product/
  - Terminal Program
    - Putty, TeraTerm, ect.

# M1 & M3 Cortex Architecture

- **Both M1 & M3 Cortex Architectures offer**
  - 32 bit processor
  - Arm Arch V6 (M1)
  - Arm Arch V7 (M3)
  - Designed for Low Power, Low Foot Print, Low Interrupt Latency
  - Communication with peripherals
  - 32 Bit hardware multiplier if desired
  - Debug Access Port – JTAG / Serial Wire access

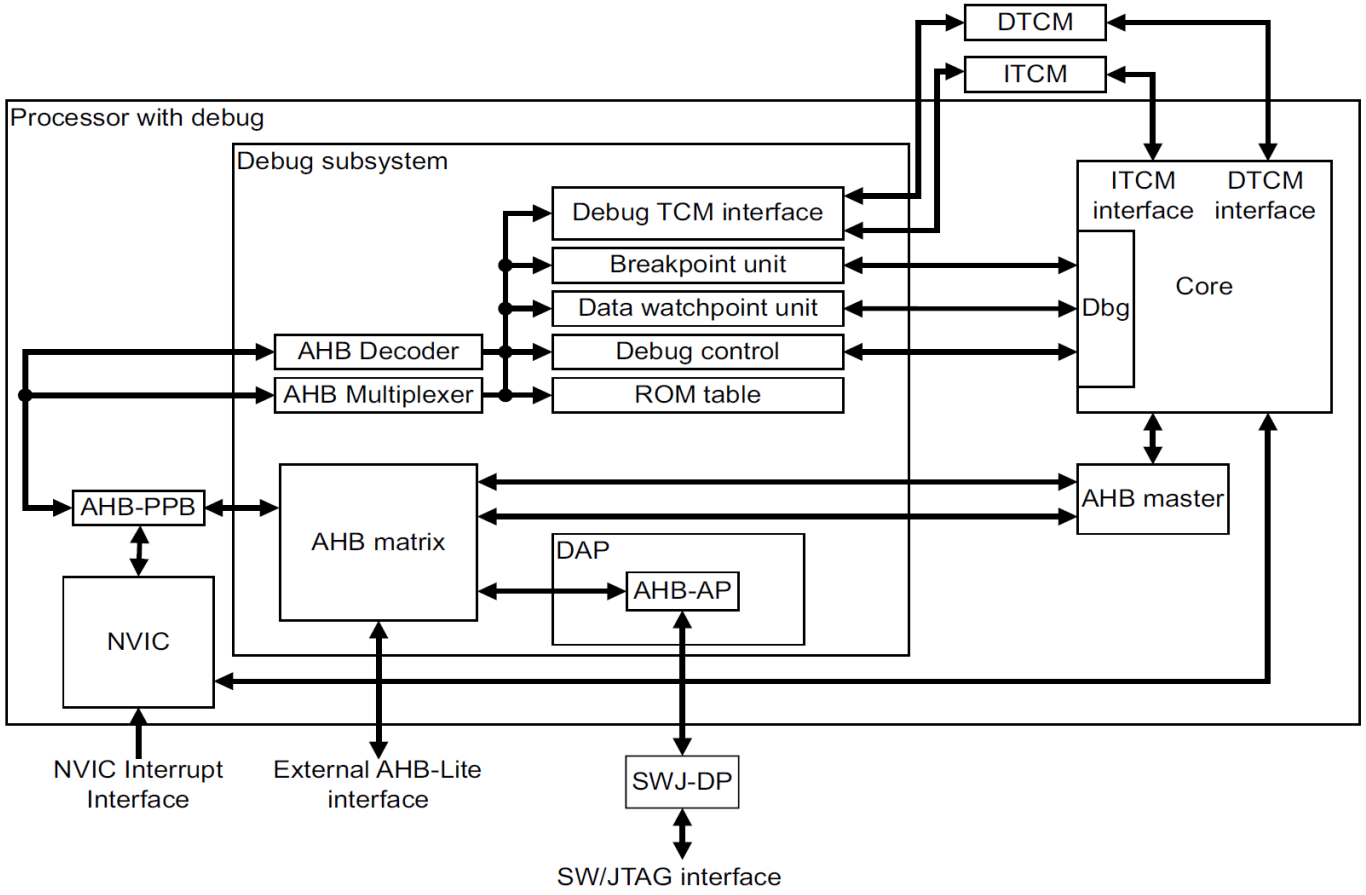
- **Forward Binary Compatibility**
  - M1 core implements a subset of the M3 instruction set and features

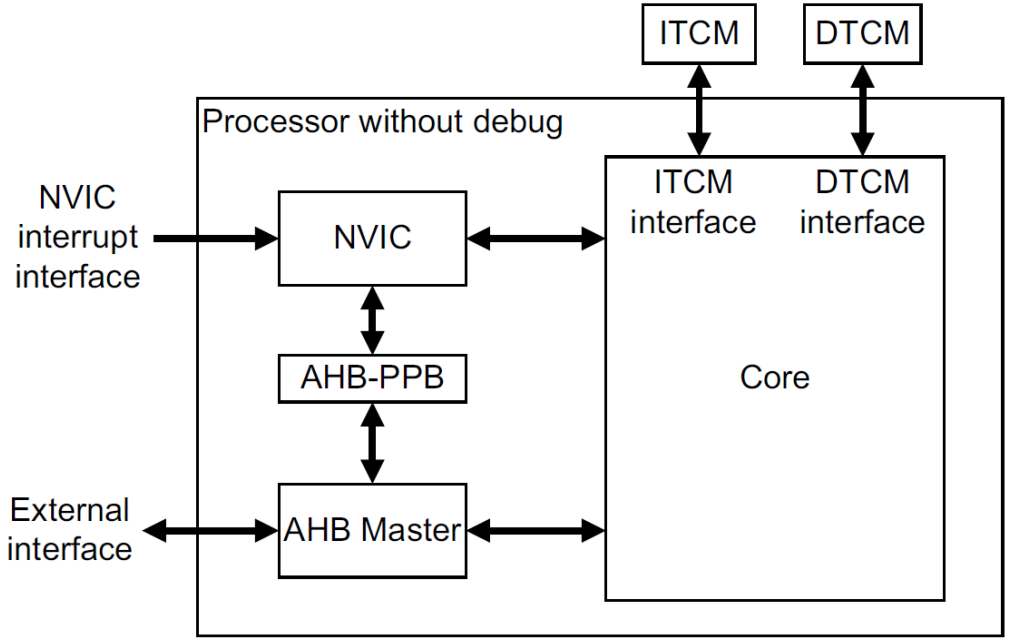
# Optional Features

- **M1 Options**
- Debug access to all memory and registers in the system, including the processor register bank when the core is halted
- Debug Access Port (DAP)
- *BreakPoint Unit* (BPU) for implementing breakpoints
- *Data Watchpoint* (DW) unit for implementing watchpoints

- **M3 Options**
- Memory Protection Unit (MPU).
- Flash Patch and Breakpoint (FPB).
- Data Watchpoint and Trace Unit (DWT).
- Instrumentation Trace Macrocell Unit (ITM).
- Embedded Trace Macrocell (ETM).
- Advanced High-performance Bus Access Port (AHB-AP).
- AHB Trace Macrocell interface (HTM interface).
- Trace Port Interface Unit (TPIU).
- Wake-up Interrupt Controller (WIC).
- Debug Port Debug Port AHB-AP interface.
- Constant AHB control.

# M1 Block Diagram



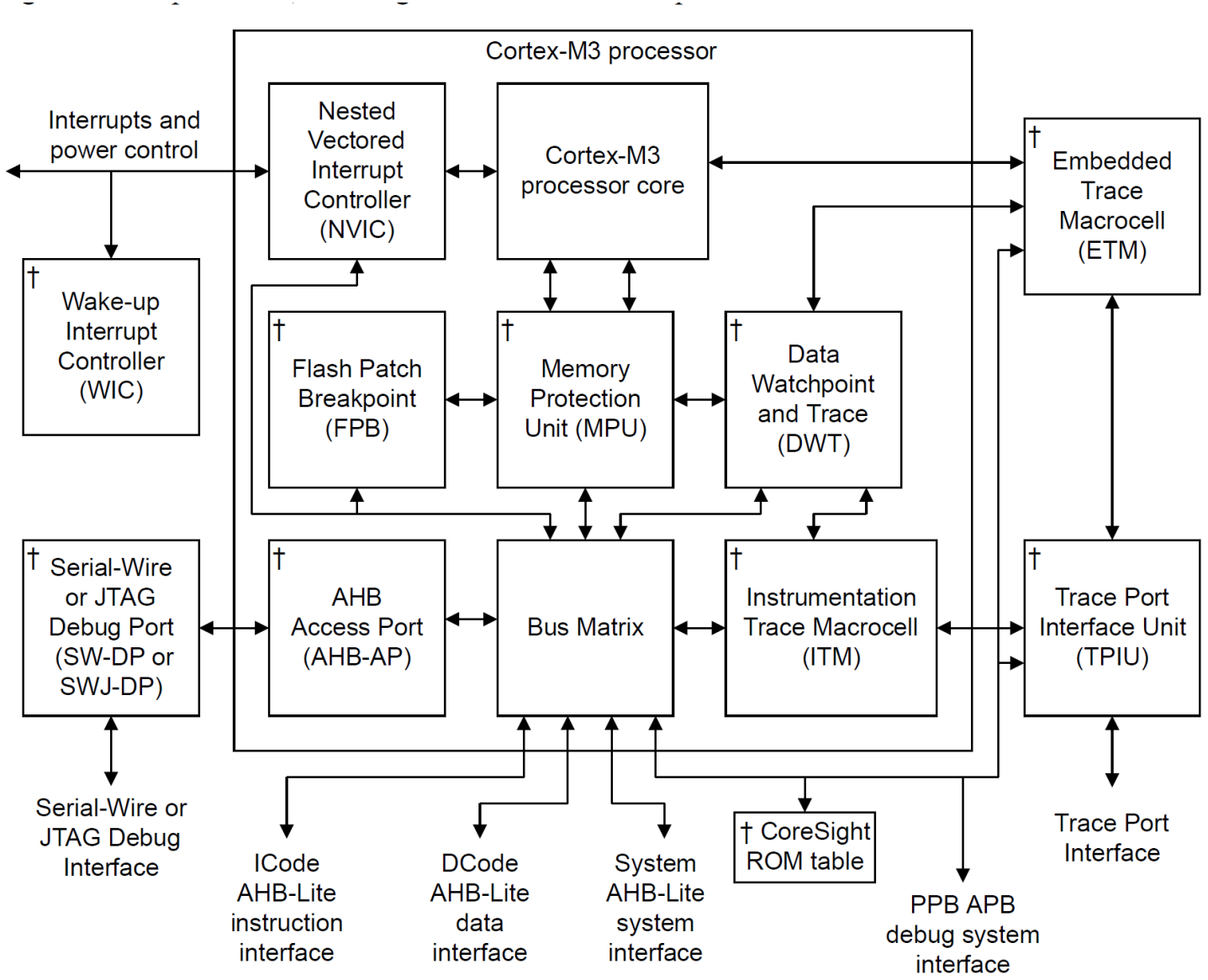








XILINX.

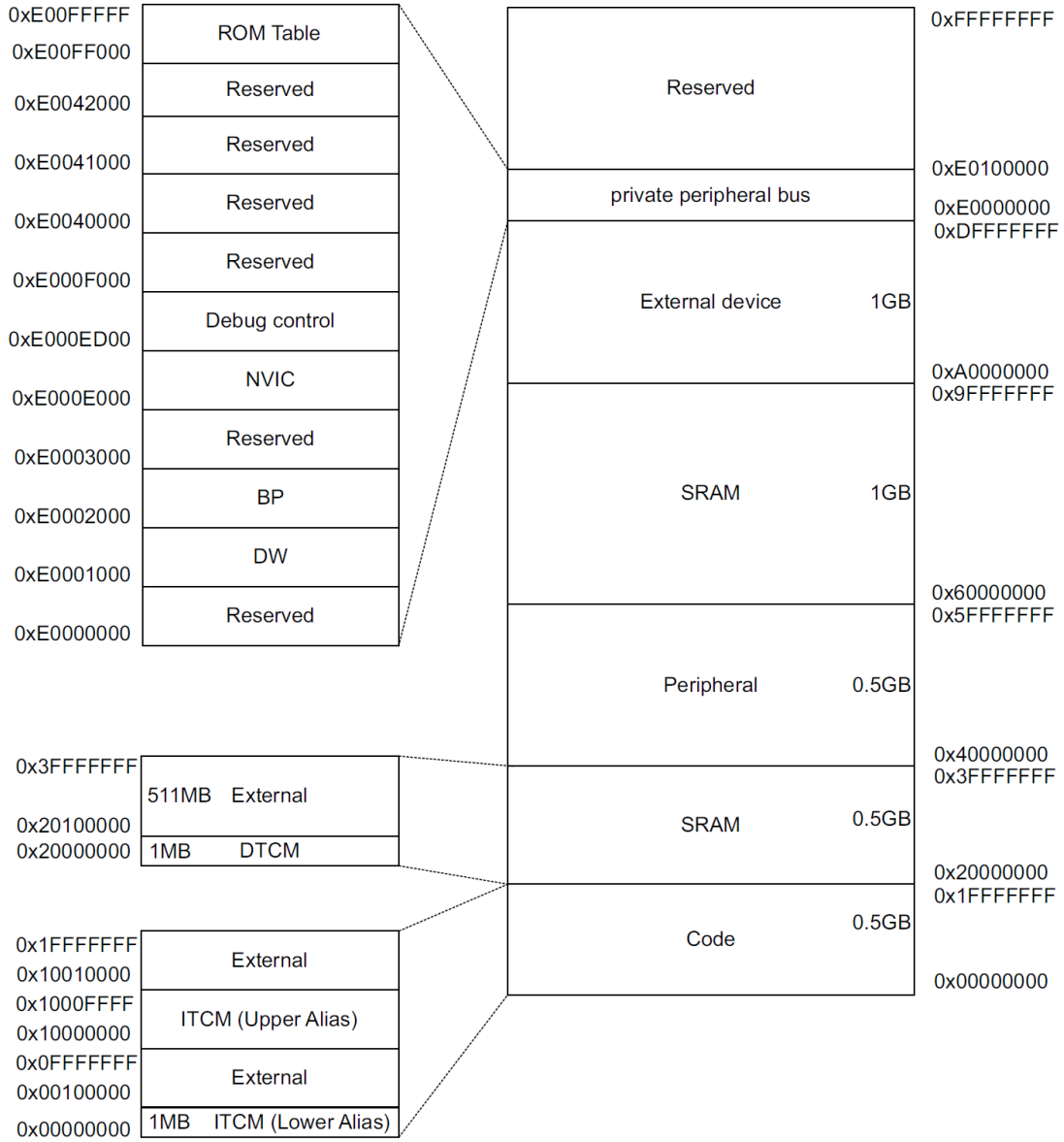
# M3 Block Diagram



# M1 Details

- **No Privilege / User mode**
- **Two modes Thread & Handler**
  - Thread Mode entered on reset
  - Handler mode result of an exception  e.g interrupt
- **Two Operating States**
  - Thumb – Normal running mode implementing Thumb & Thumb-2 Instructions
  - Debug – Halted for debugging
- **Registers**
  - 13 general-purpose registers, R0-R12
  - Stack Pointer (SP) (SP, R13) and banked register aliases, SP_process and SP_main
  - Link Register (LR, R14)
  - Program Counter (PC, R15)
  - Program status registers, xPSR.

# M1 Memory Map








XILINX.

# M1 Word Alignment

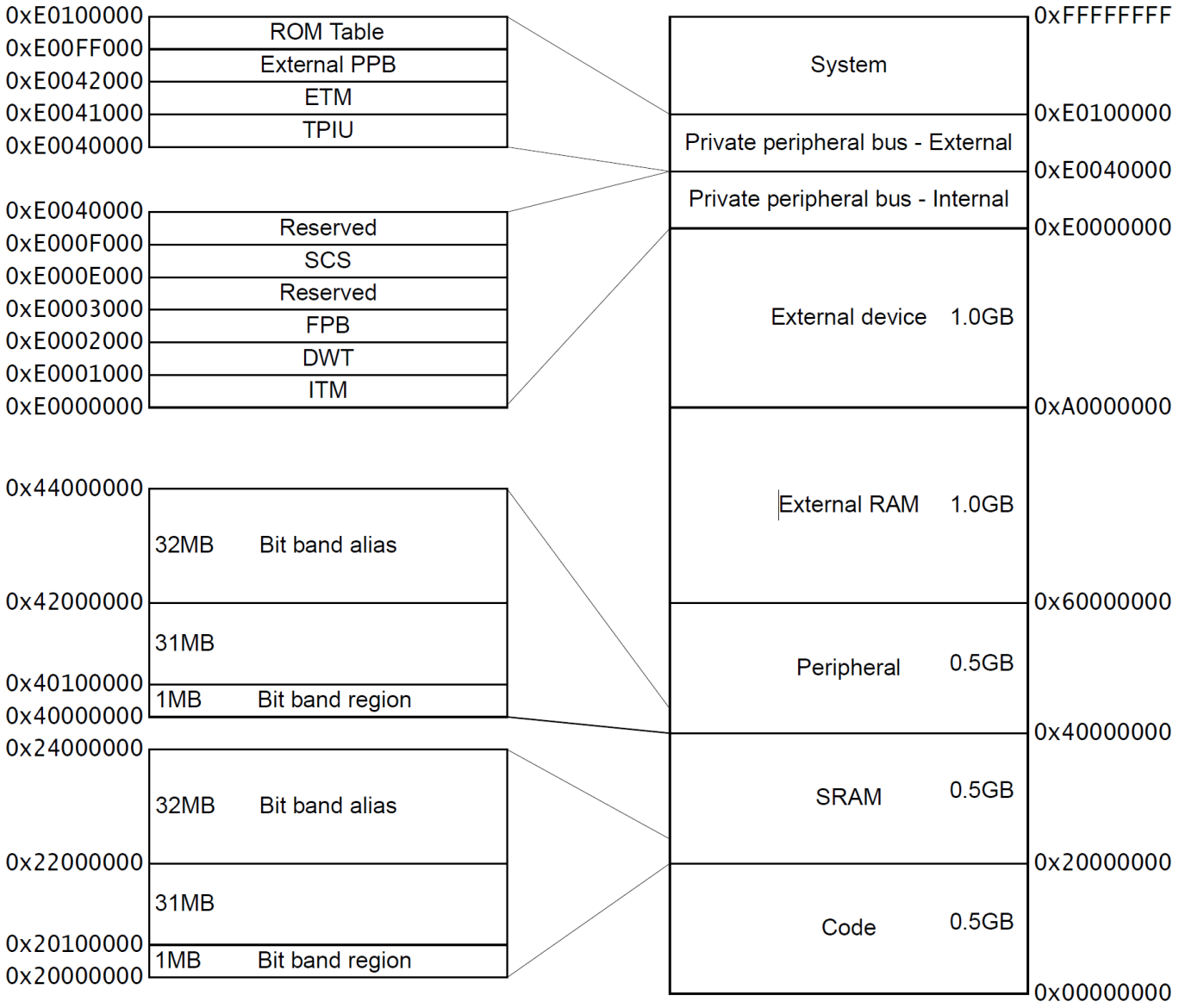
> **Memory is viewed as a linear collection of bytes in ascending order**

| HSIZE | HADDR[1:0] | DATA[31:24] | DATA[23:16] | DATA[15:8] | DATA[7:0] |
|---|---|---|---|---|---|
| Word | 0 | x | x | x | x |
| Halfword | 0 | - | - | x | x |
| Halfword | 2 | x | x | - | - |
| Byte | 0 | - | - | - | x |
| Byte | 1 | - | - | x | - |
| Byte | 2 | - | x | - | - |
| Byte | 3 | x | - | - | - |

# M3 Details

- **User level of Privilege or Un-Priviledge – Processor can restrict access depending on level**
- **Two modes Thread & Handler**
  - Thread Mode entered on reset
  - Handler mode result of an exception  e.g interrupt
- **Two Operating States**
  - Thumb – Normal running mode implementing Thumb & Thumb-2 Instructions
  - Debug – Halted for debugging
- **Registers**
  - 13 general-purpose registers, R0-R12.
  - *Stack Pointer* (SP), R13 alias of banked registers, SP_process and SP_main.
  - *Link Register* (LR), R14.
  - *Program Counter* (PC), R15
  - Special-purpose *Program Status Registers*, (*x*PSR).

# M3 Memory Map



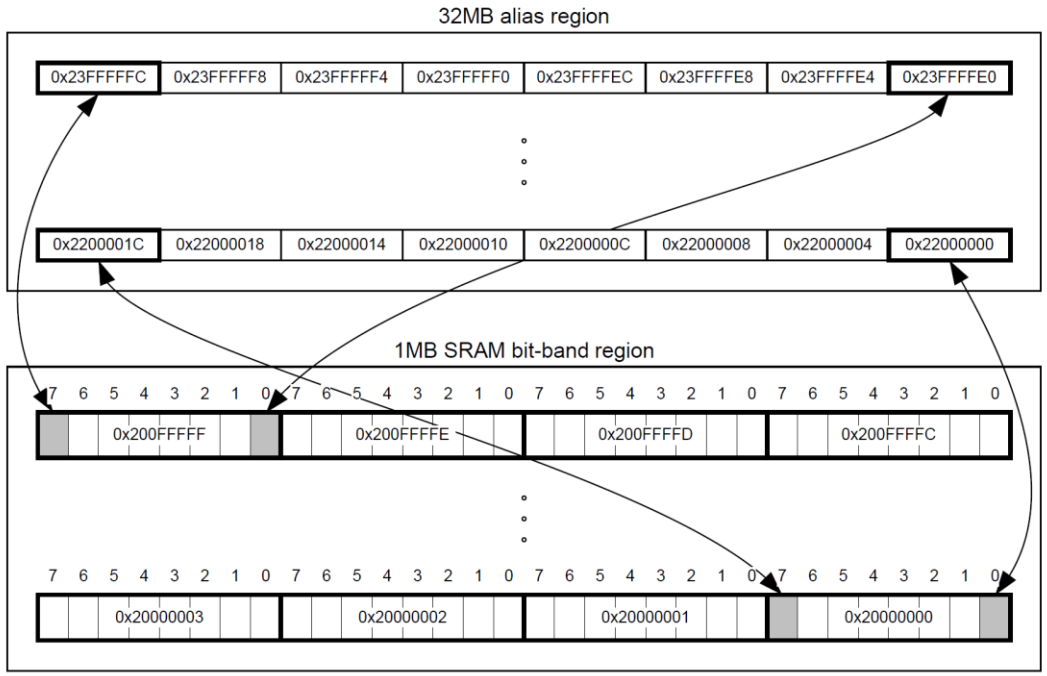





XILINX.

# M3 Bit Banding

- Bit-banding enables every individual bit in the bit-banding region to be directly accessible from a word aligned address using a single LDR instruction. It also enables individual bits to be toggled without performing a read-modify-write sequence of instructions.
- The processor memory map includes two bit-band regions.








XILINX.

# What is Thumb / Thumb 2

- **Thumb / Thumb 2 is critical technology in achieving the exceptional code densities**

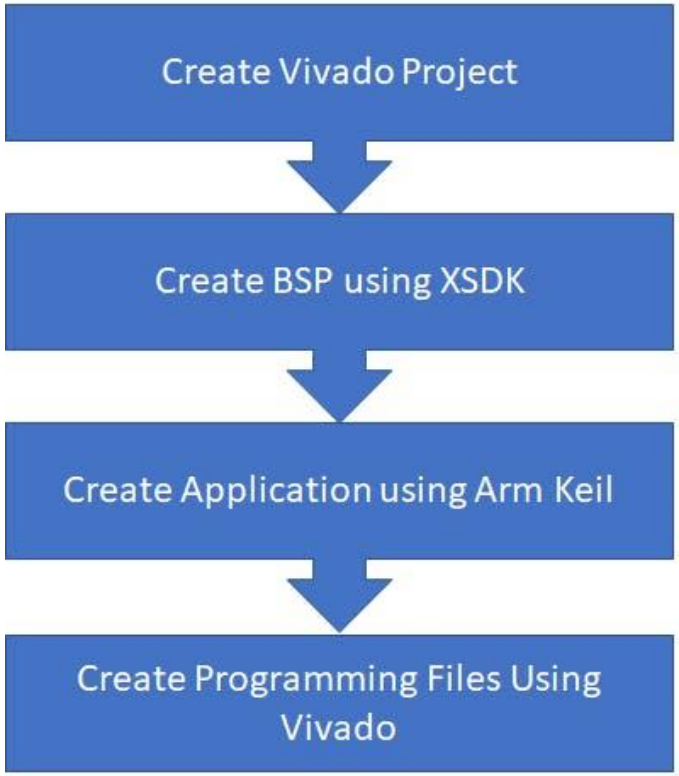
- **Thumb is  16 bit encoded subset of the Arm Instruction set**
  - Some limitations – Branches conditional and limits access to half of all GP registers
  - Better memory density even is additional instructions are needed

- **Thumb 2 extends the 16 bit encoded subset to 32 bits introducing a variable length instruction set**
  - Target was to provide code densities like Thumb but with performance of 32 bit instructions set
  - Extends Thumb with bit filed manipulation, table branches and conditional execution

# Development Flow

# Development Flow

- **Creating a M1 or M3 Cortex Implementation requires the following development flow**




Using Vivado we create the FPGA design, including the Cortex M3 Processor and connected peripherals

Once the hardware is created, we export the hardware definition file. This hardware definition file is used by Xilinx SDK to create the Board Support Package. This provides the API and drivers for the processor configuration.

Using Arm Keil, we use the BSP and create the application itself.

The ELF file from Arm Keil is merged with the FPGA configuration file using Vivado. This creates a BIT file which can be used to program the FPGA over JTAG and a MCS file which can be loaded into the configuration memory

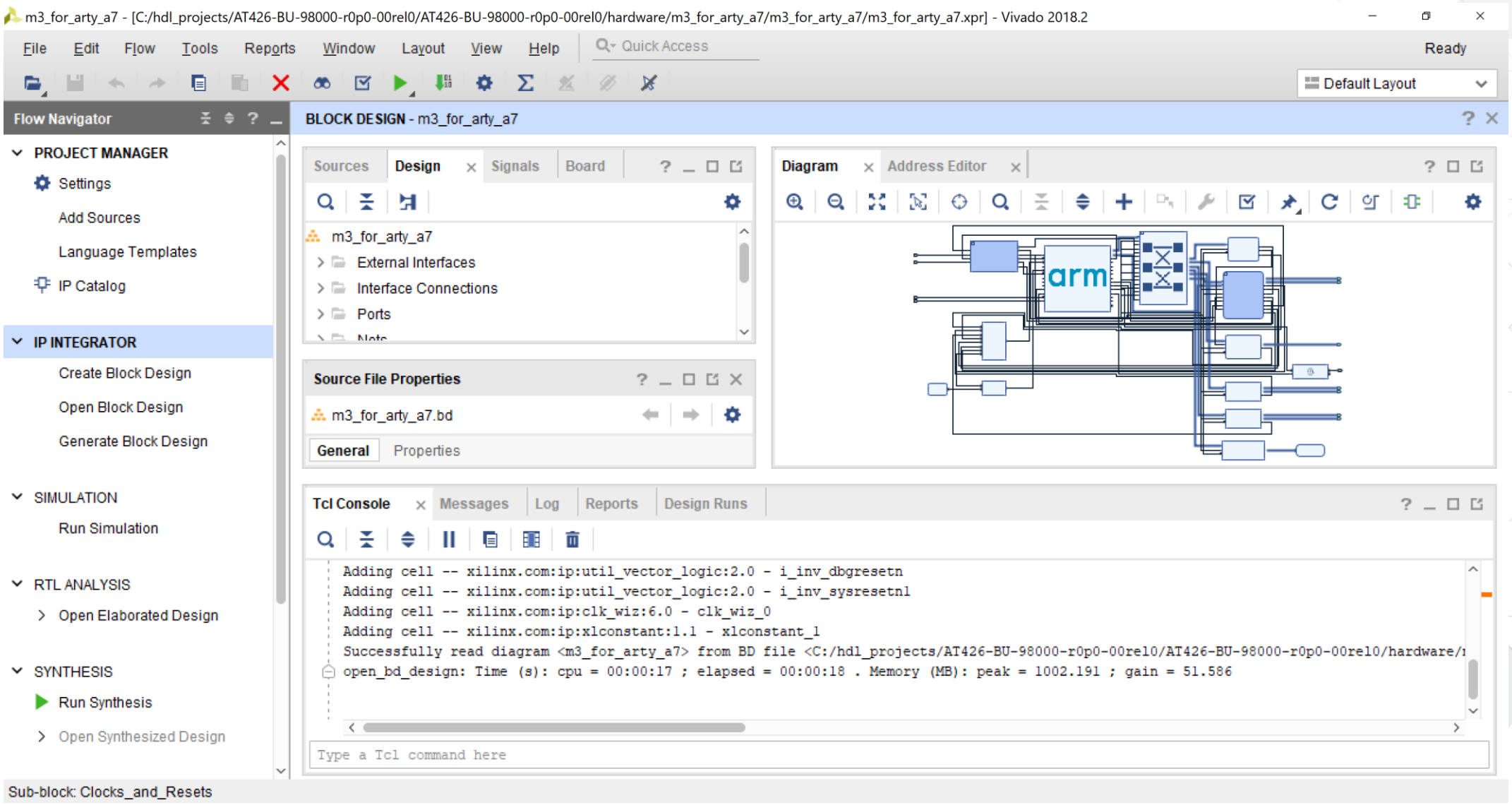
# Vivado

- **Vivado – Xilinx design tool which is used to capture the FPGA**

- **Design entry can use**
  - Register Transfer Level Language e.g. Verilog or VHDL
  - Vivado High Level Synthesis Generated IP
  - IP Integrator – Allows leverage of Xilinx and Partner IP in graphical environment

- **Using IP Integrator we can implement M1 and M3 designs without writing a line of code**

# Vivado IP Integrator



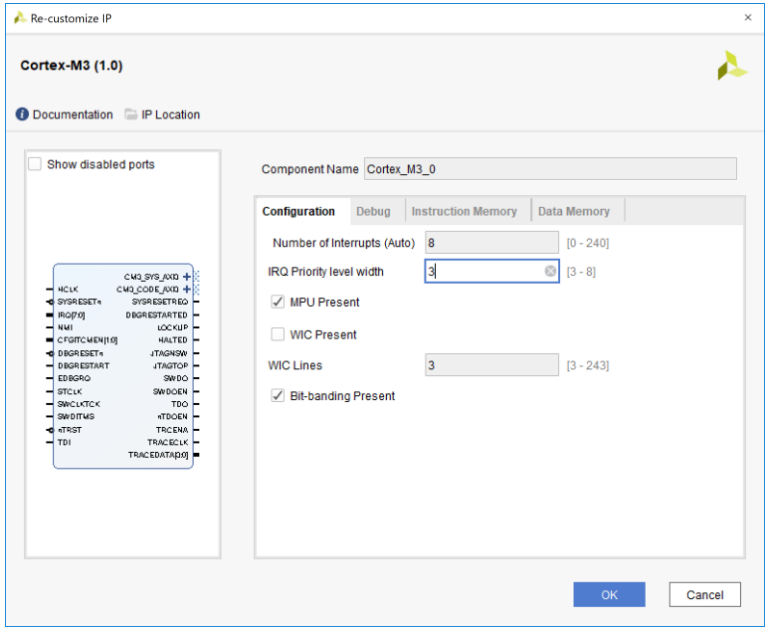


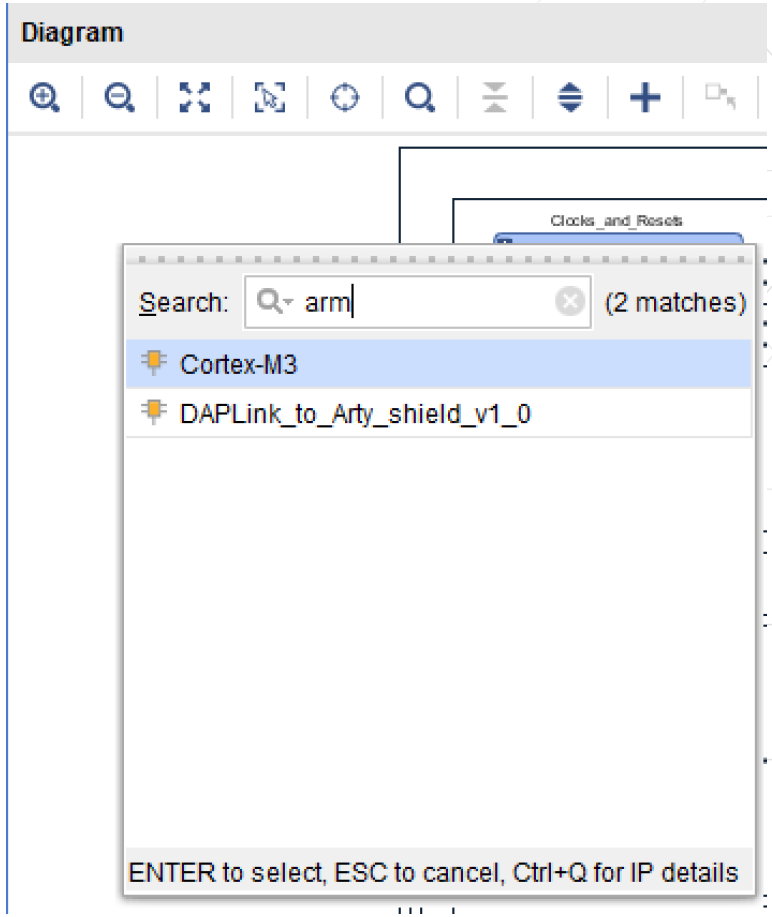



XILINX.

# Adding in the Processor

- Once the processor has been downloaded it can be added to the IP Integrator design just like any other IP
- Processor and DAP Link IP cores are provided
- Once added to the IPI the processor can be customized by double clicking on it.



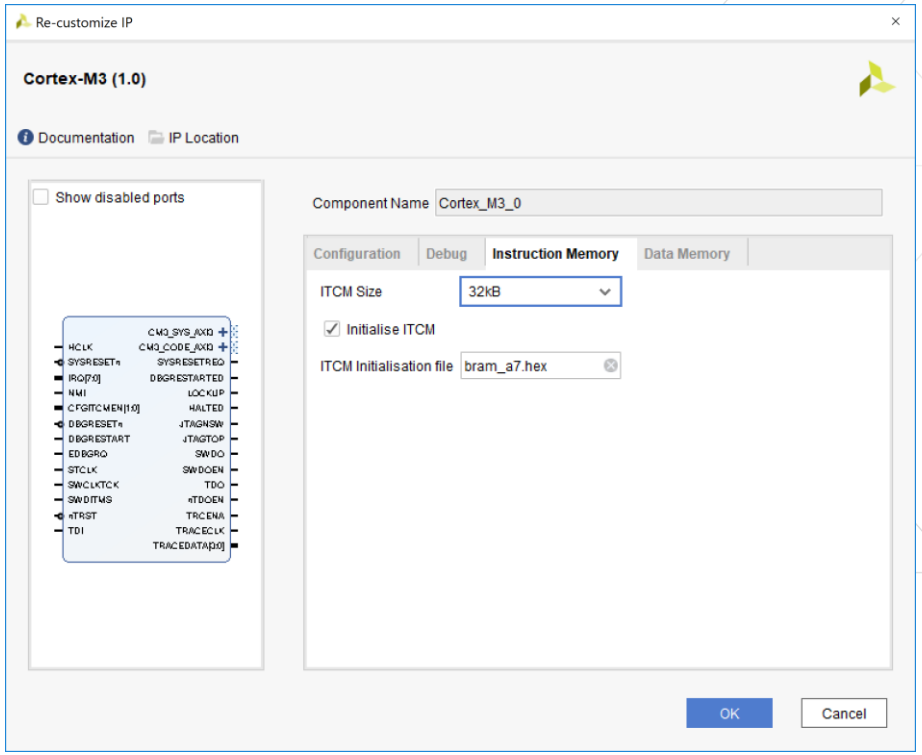





XILINX.

# Memory Size

- Tightly Coupled Instruction and Data Memory is implemented in FPGA Block RAM
- This can be a driving factor on your FPGA Choice
- Instruction Memory Size can range from 8 KB (XC7S6) to 1MB  (XC7A200T)
- Data Memory Size can range from 2KB to1MB



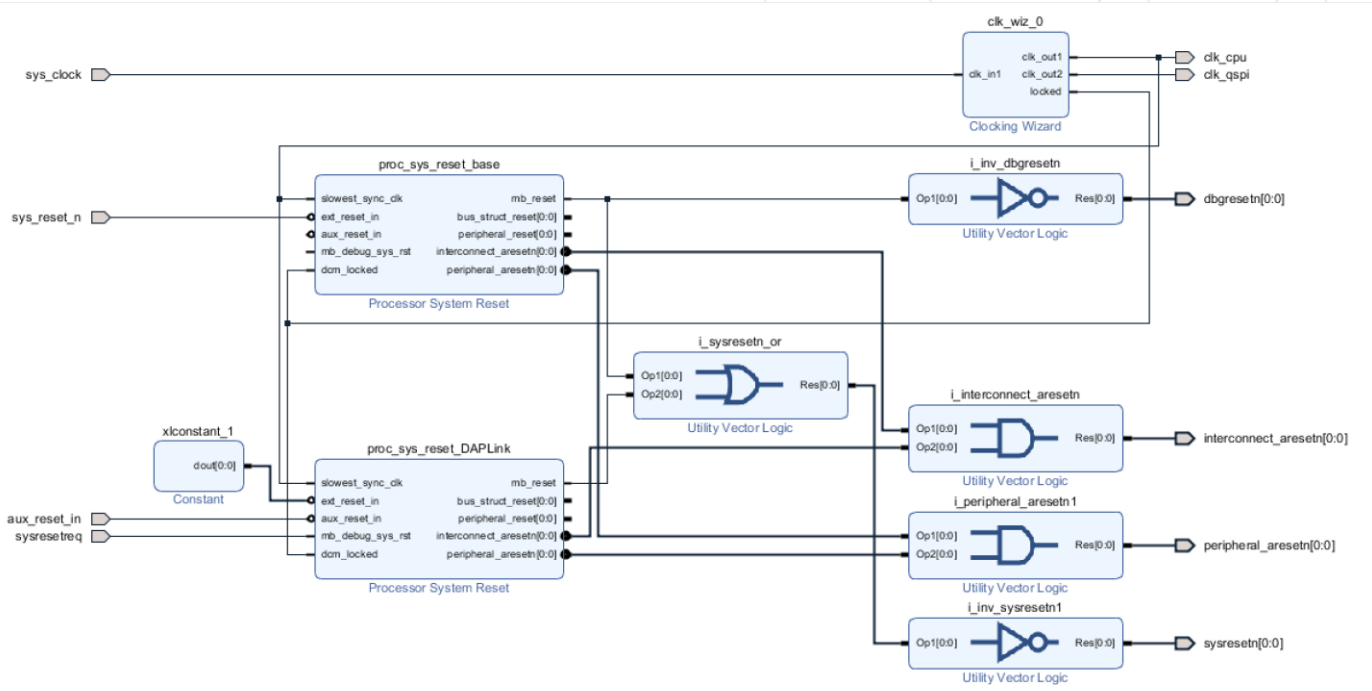





XILINX.

# Clocking & Reset

- **Single Clock – FPGA Fabrice selected will determine maximum frequency**
- **Typical implementations will achieve timing closure easily with 50 MHz and 100MHz**
- **Resets are a little more complex**
  - System reset is needed for power on reset
  - Debug Reset is needed for debugging







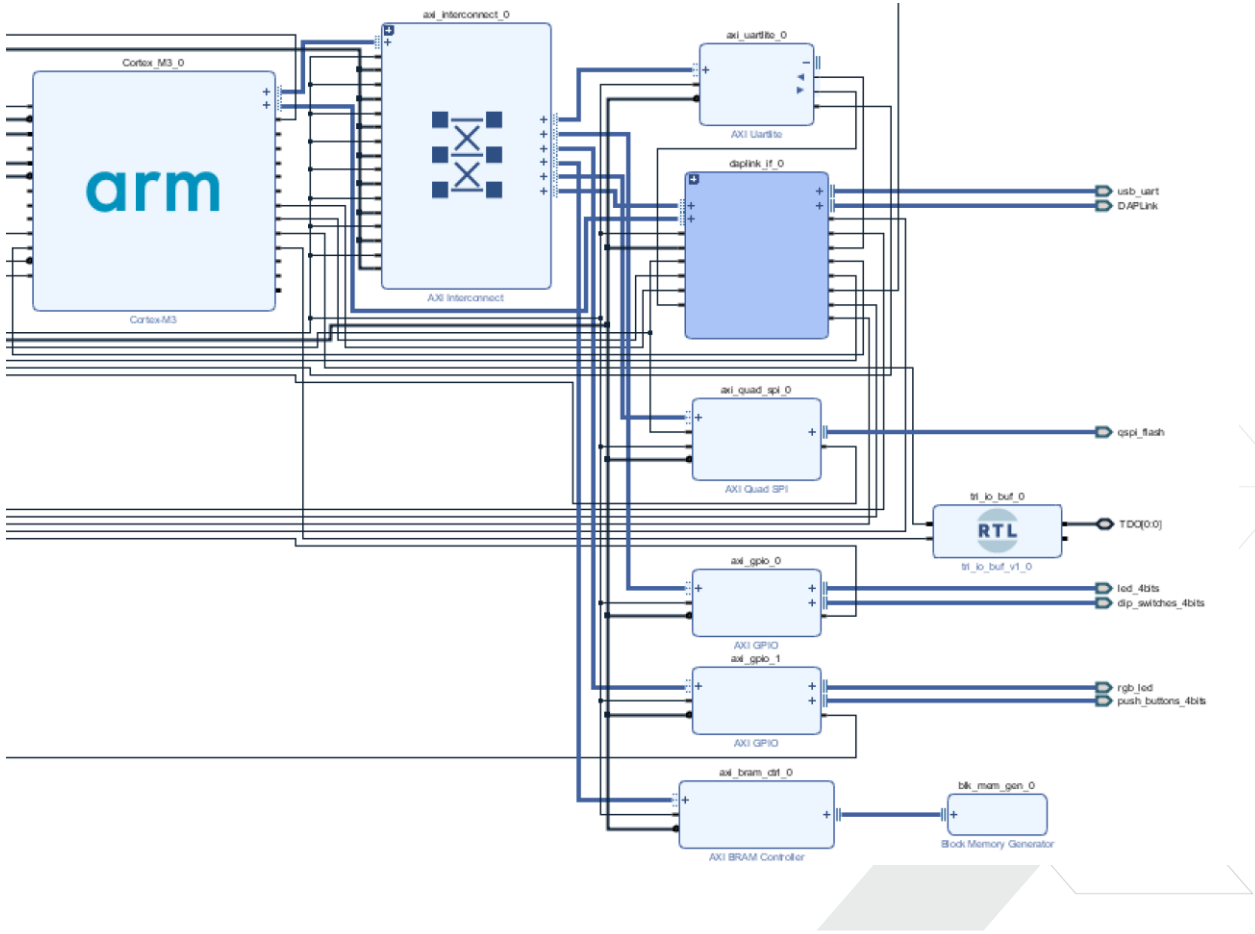

XILINX.

# AXI Interconnect

- **Peripherals are added into the M1 / M3 Cortex system using the AXI Interface**
- **AXI = Arm eXtensible Interface**
- **Three Flavors of AXI used in Xilinx FPGA designs**
  - AXI Full – Memory Mapped Full Duplex interface with burst access. This is used for transferring large data quantities e.g. Direct Memory Access Transfers
  - AXI Stream – Uni-drectional stream of data from a master to a slave, as a continuous stream of data no addressing is needed.
  - AXI Lite – Like AXI Full this is a memory mapped full duplex interface, however it does not support bursts being limited to a single beat. This is meant for control and configuration etc.
- **AXI Lite is the interface used by the Arm M1 & M3 devices to interface with peripherals**

# AXI Interconnect

- AXI is a point to point protocol
- Communication is always between a master and a slave
- For a single master to communicate with multiple slaves a Cross Bar is required in the form of AXI Interconnect or Smart Interconnect.








XILINX.

# What do we need from Vivado

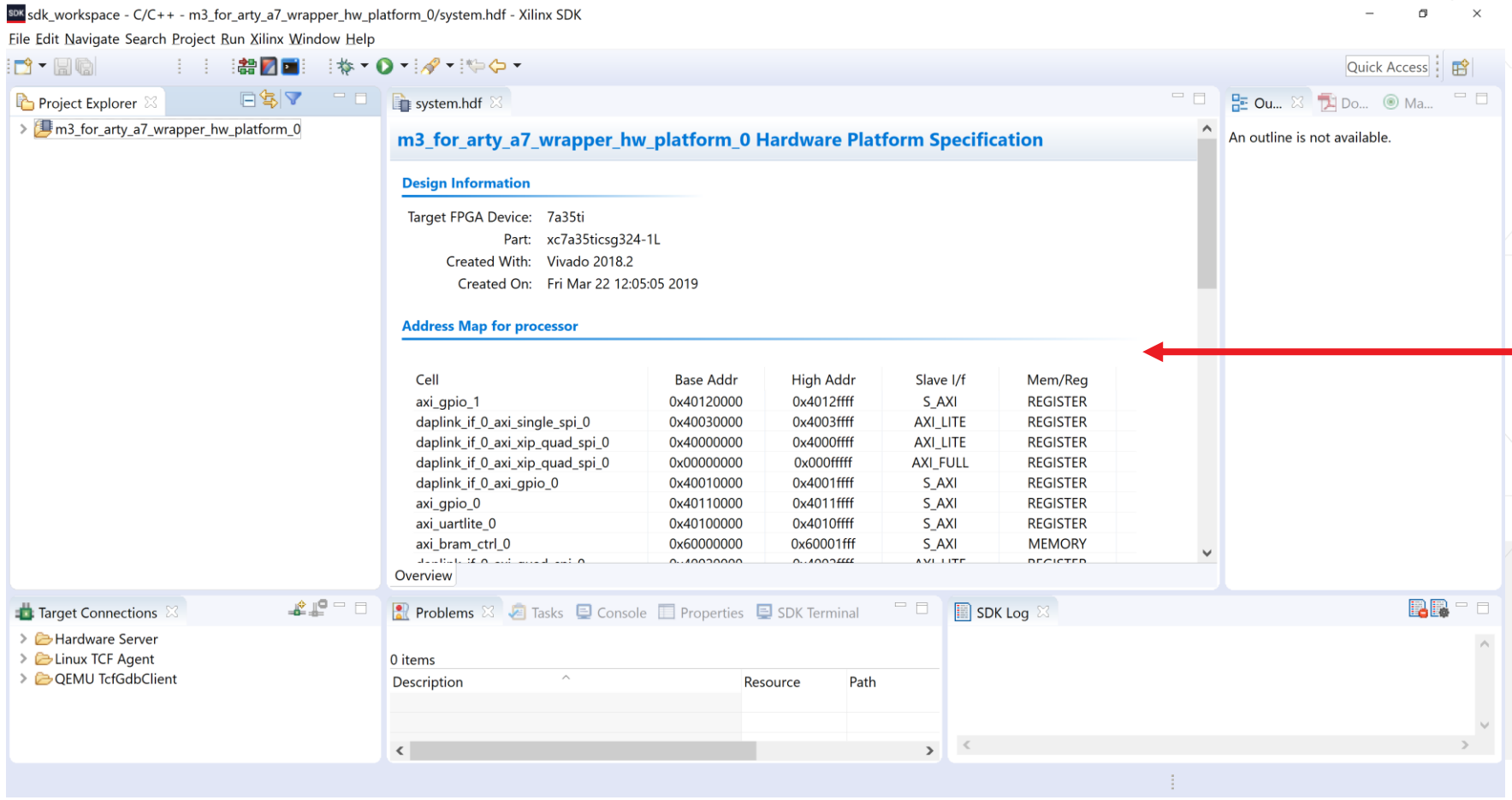
- **Bit File – Generated once your IPI design is completed**
- **HDF - Defines the Peripherals in your design and crucially how they are mapped into the processor memory**
- **MMI File – Describes the Block Ram Memory locations to enable SW application to be updated without the need for the entire FPGA to be recompiled.**

# Lab One
# Exploring the Architecture (20 Minutes)

# Xilinx SDK

- The design we created in Vivado will contain not only the Cortex M1 or M3 but also a number of Xilinx, Partner and Third Party IP.
- To make working with this IP easy we need to create a Board Support Package (BSP)
- This BSP will contain a number of drivers which considerably eases working with the peripherals in the design

# SDK Overview




Projects →

← Outline

← Working Area

Connections →

↑ Outputs

↑ Log

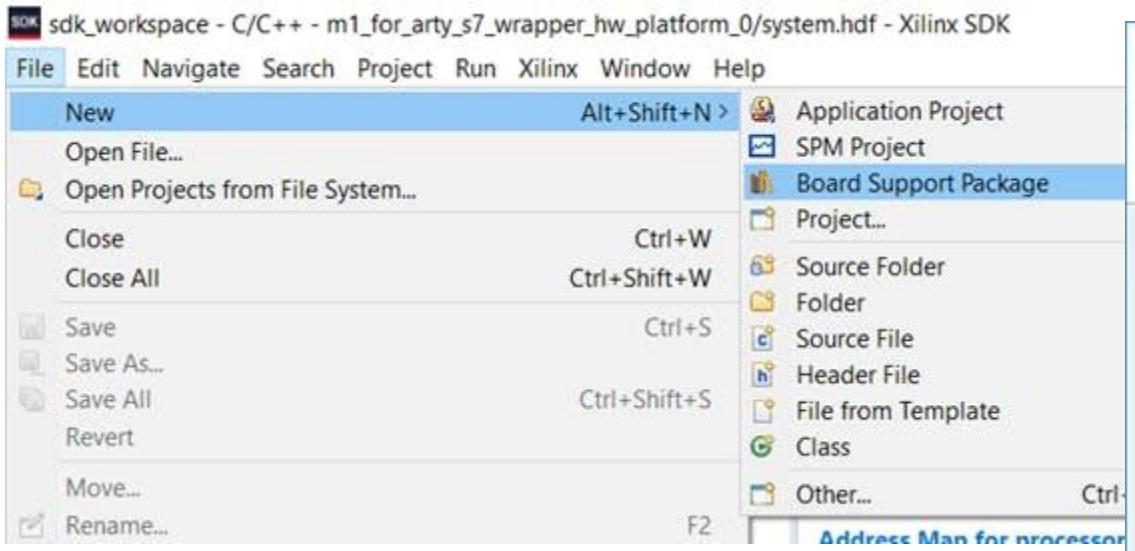


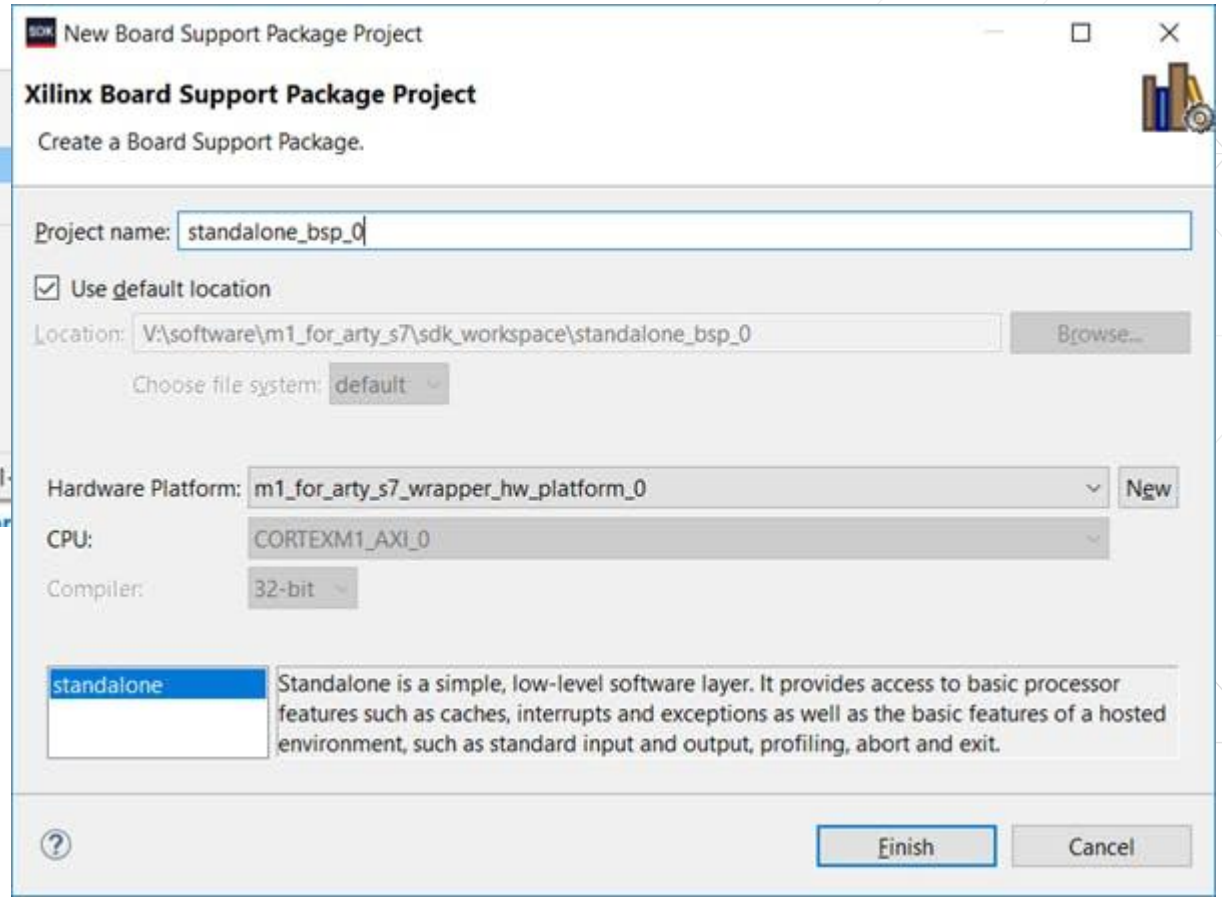


XILINX.

# Creating a BSP

- **SDK is used to create the BSP so we can access Xilinx, Partner and Third Party IP**



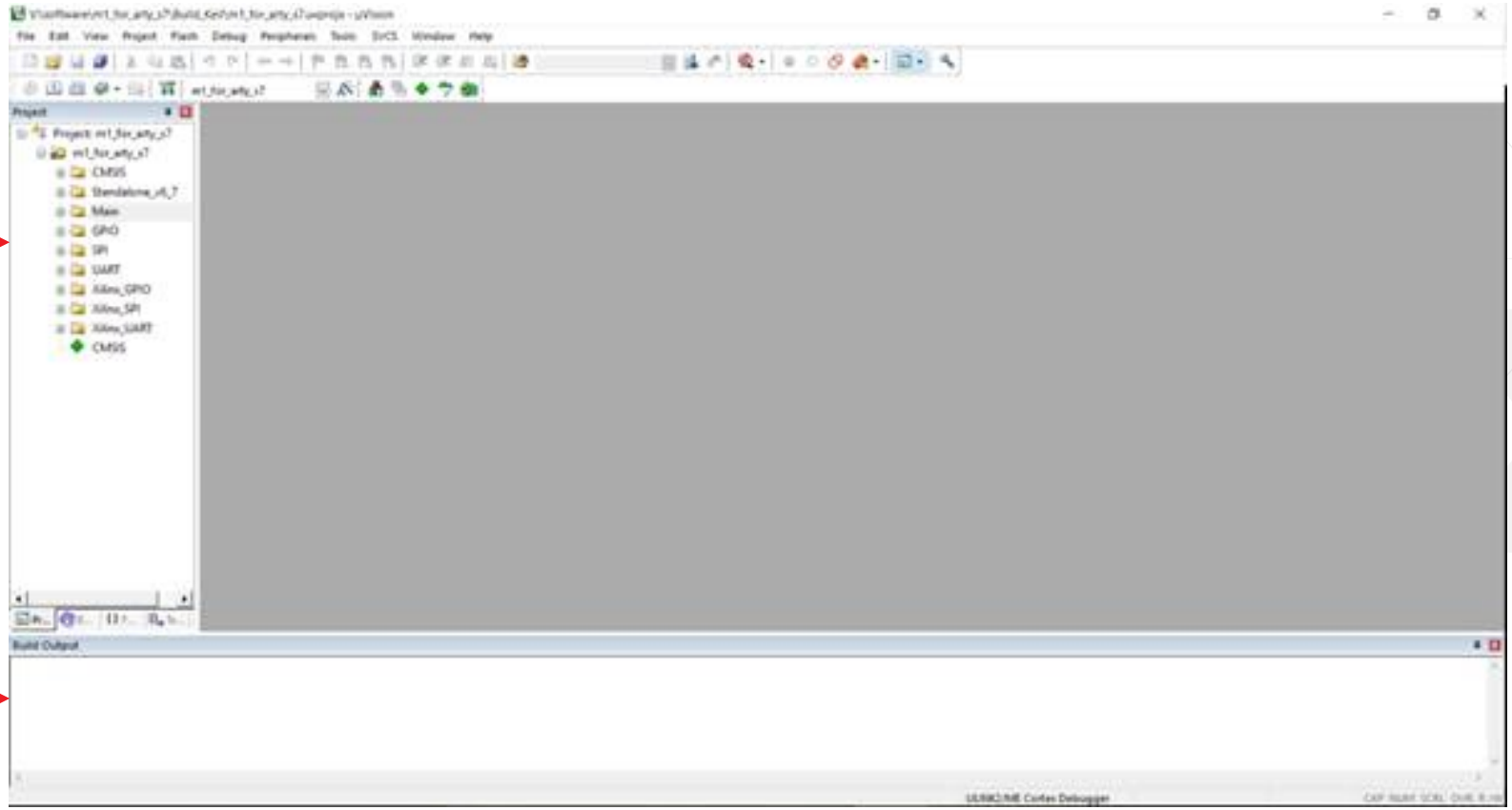





XILINX.

# Arm Keil

- **Development of the main application is performed in Arm Keil**
- **Using Xilinx IP from the BSP we need to include the c source**



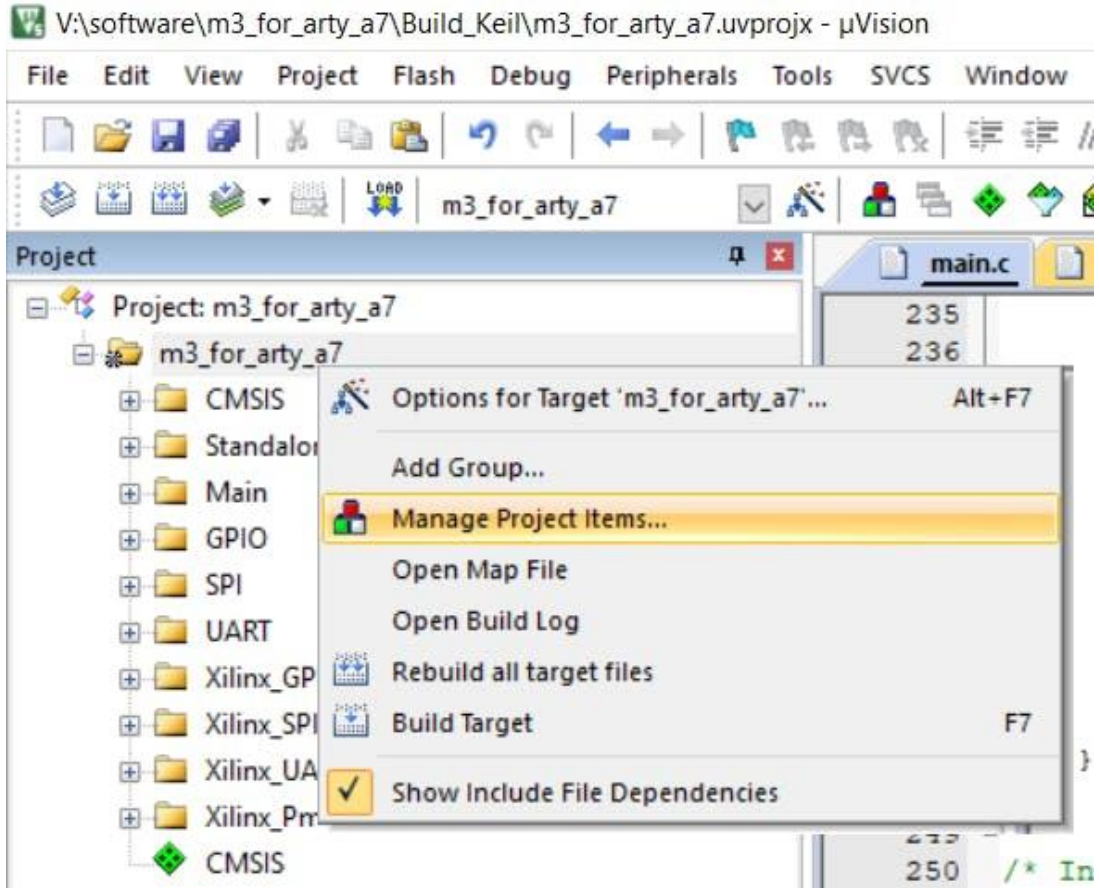


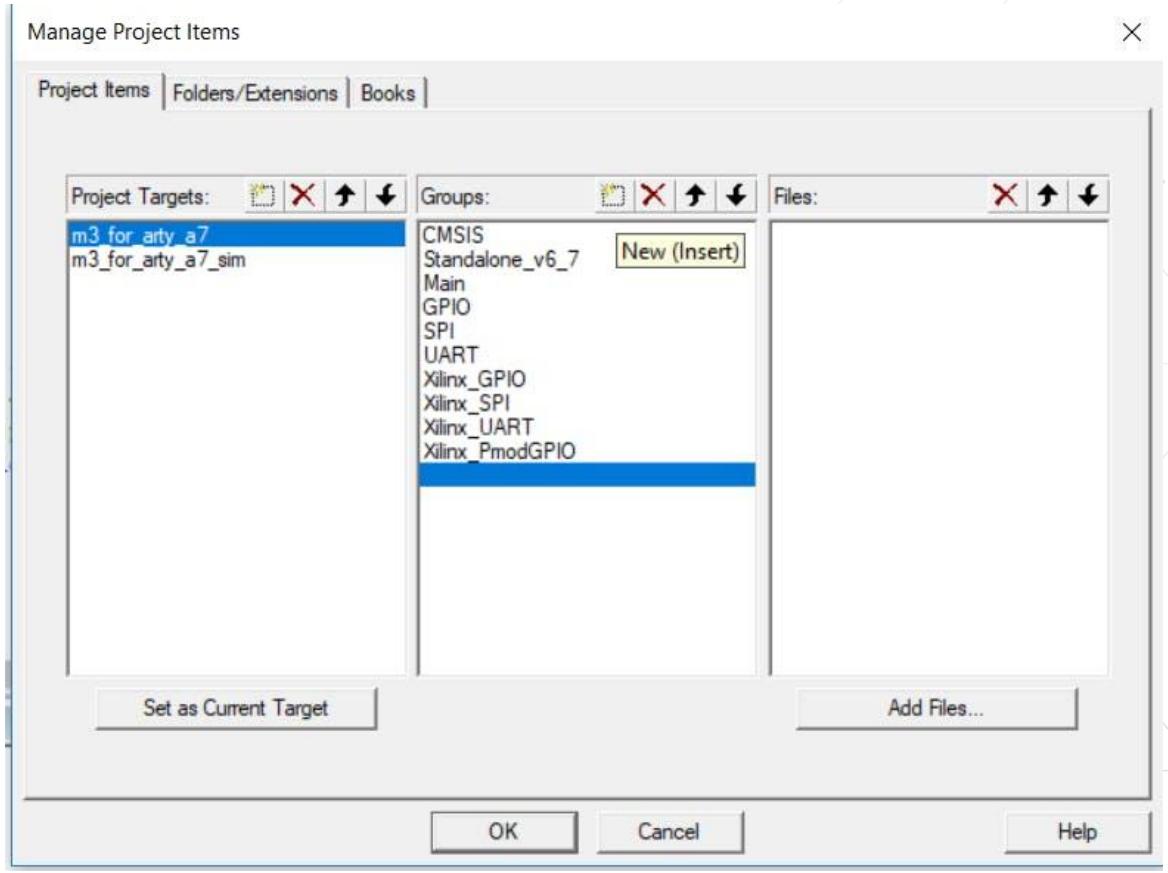



XILINX.

# Arm Keil

- Add in Xilinx BSP code using the manage project items



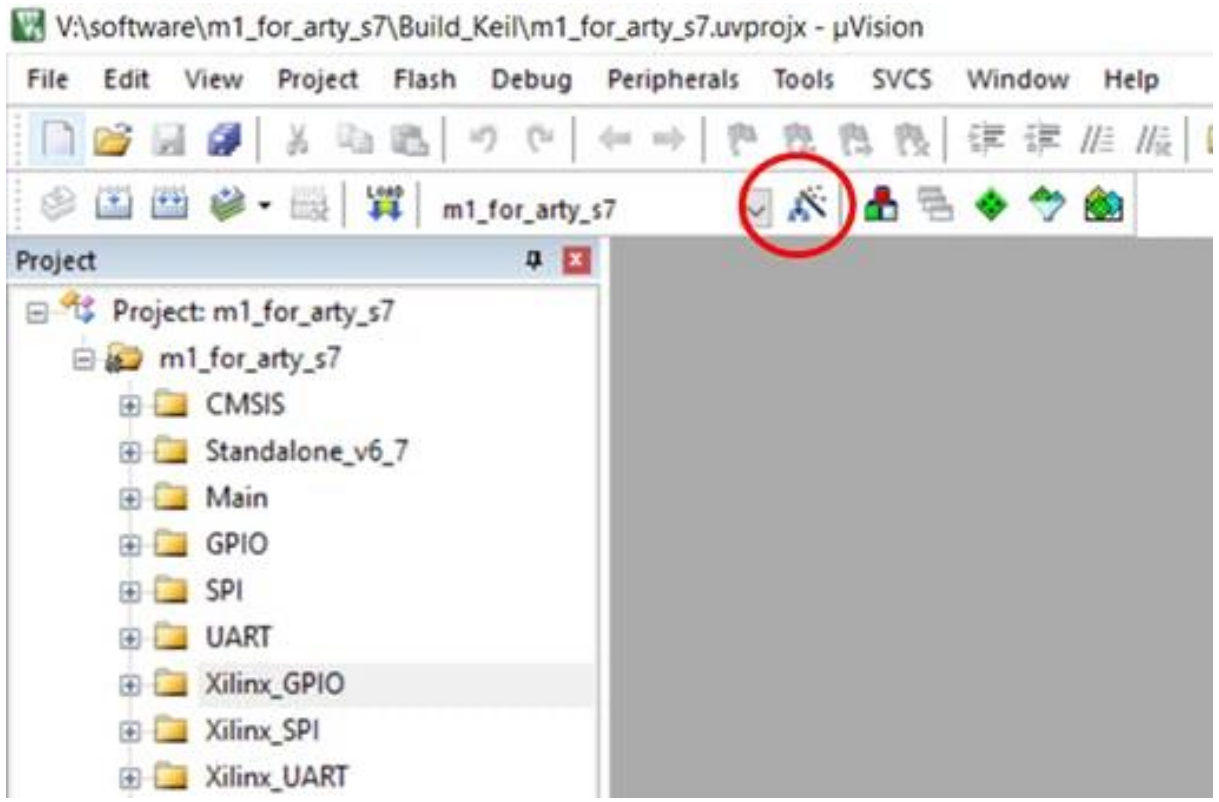


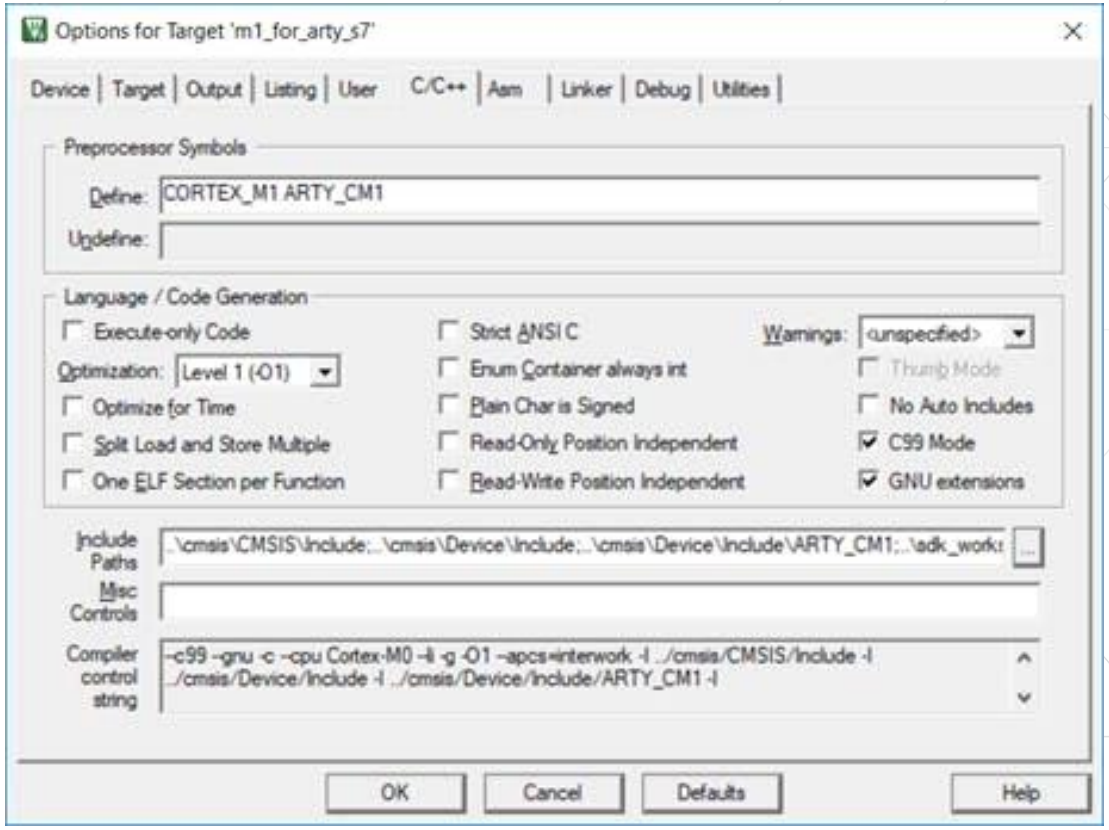



XILINX.

# Arm Keil Options

- **Compiler Options can be used for optimization, GNU Extensions, C99 compatibility**



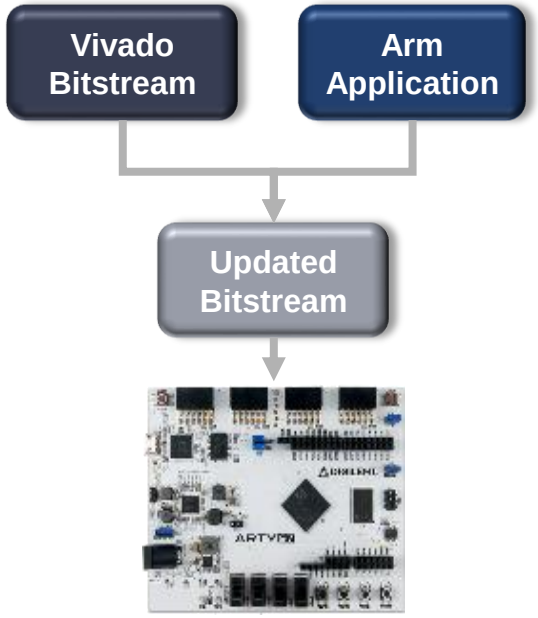





XILINX.

# Generating & Downloading the bit file

- Programming flow is to fuse the Arm Keil Application with the Vivado Bit file
- Possible as the M1/M3 application is stored in BRAM memory
- BRAM contents can be updated without the need to re implement the design
- Reference ARM design includes all of the necessary files

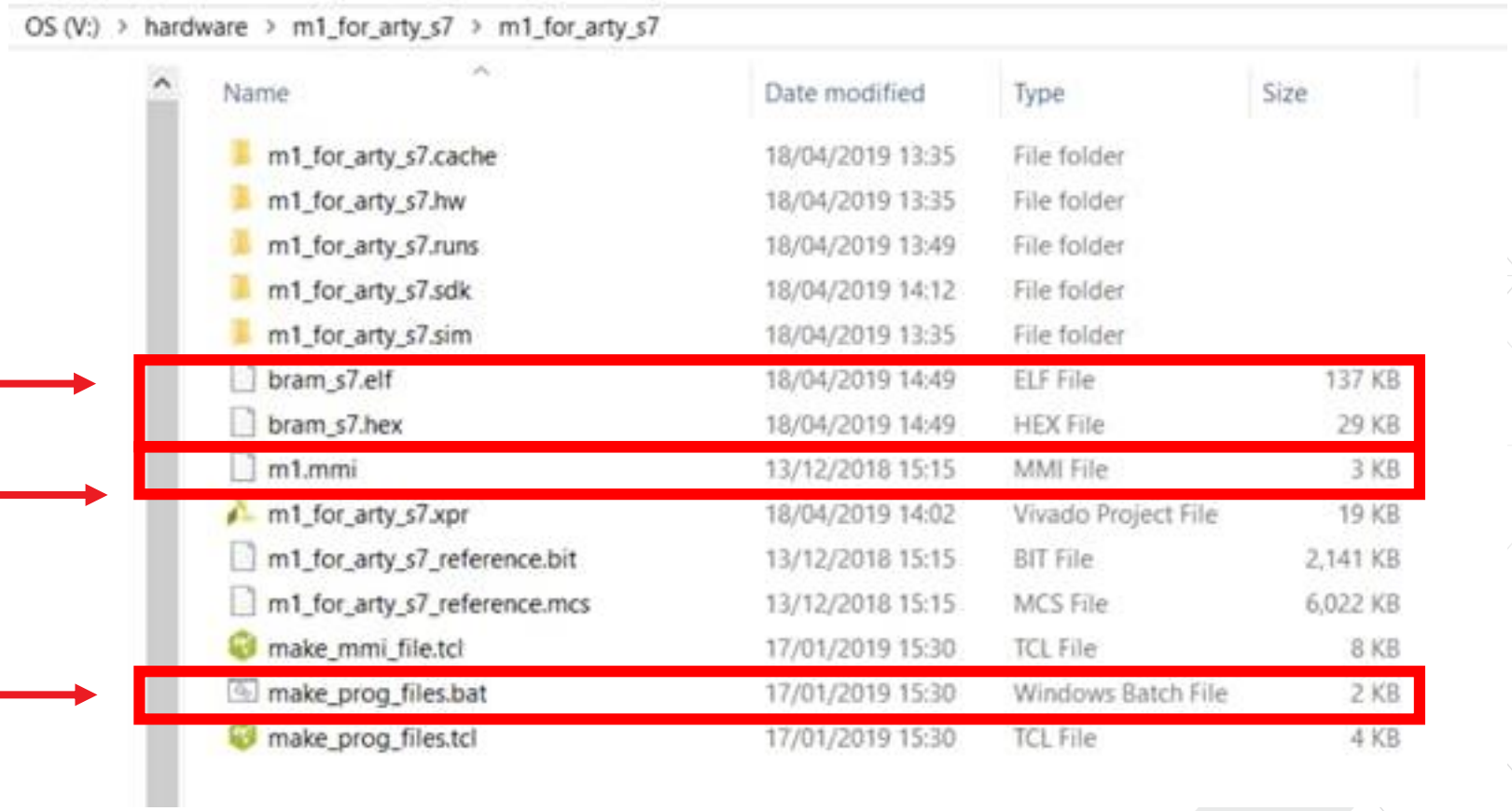


# Generating the Bit Stream

ELF and Hex files created by Arm Keil

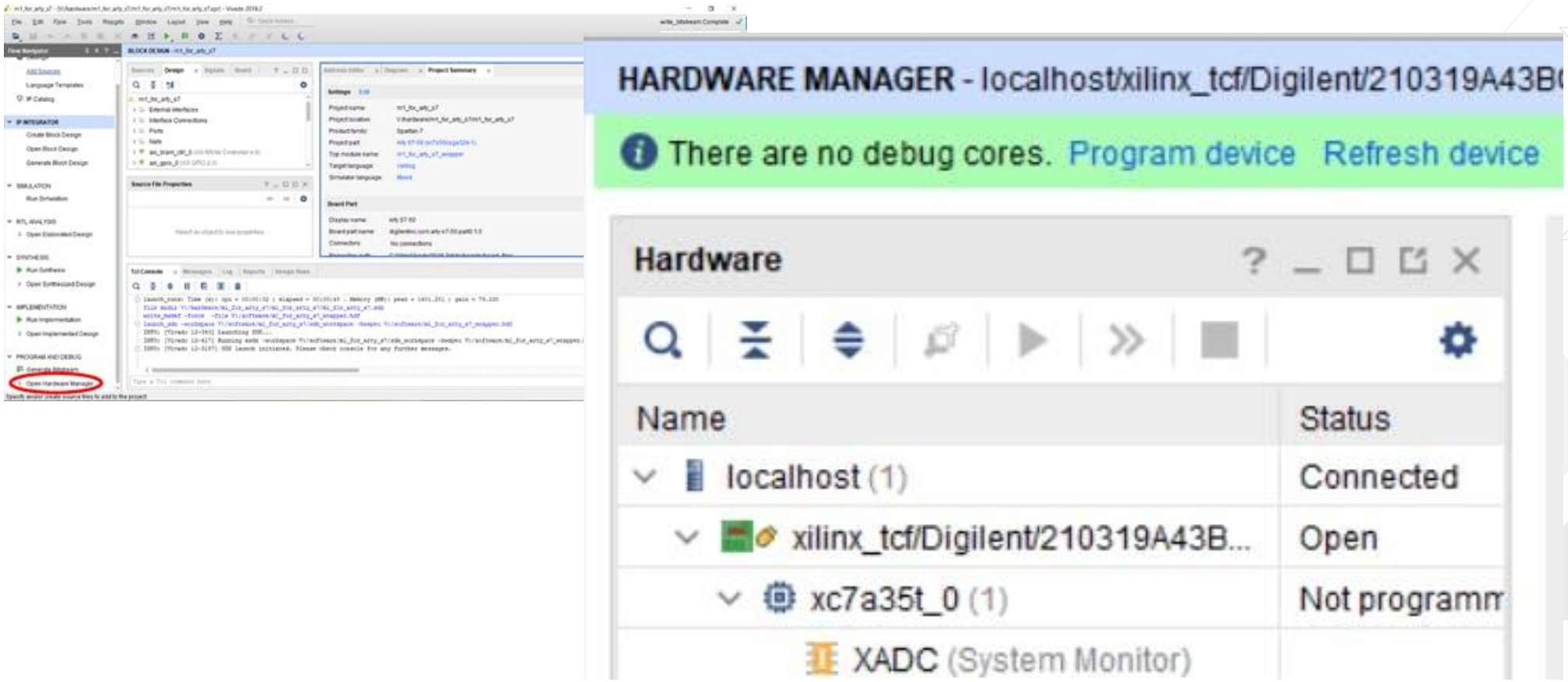
MMI FIle

BAT Files to create the programming files



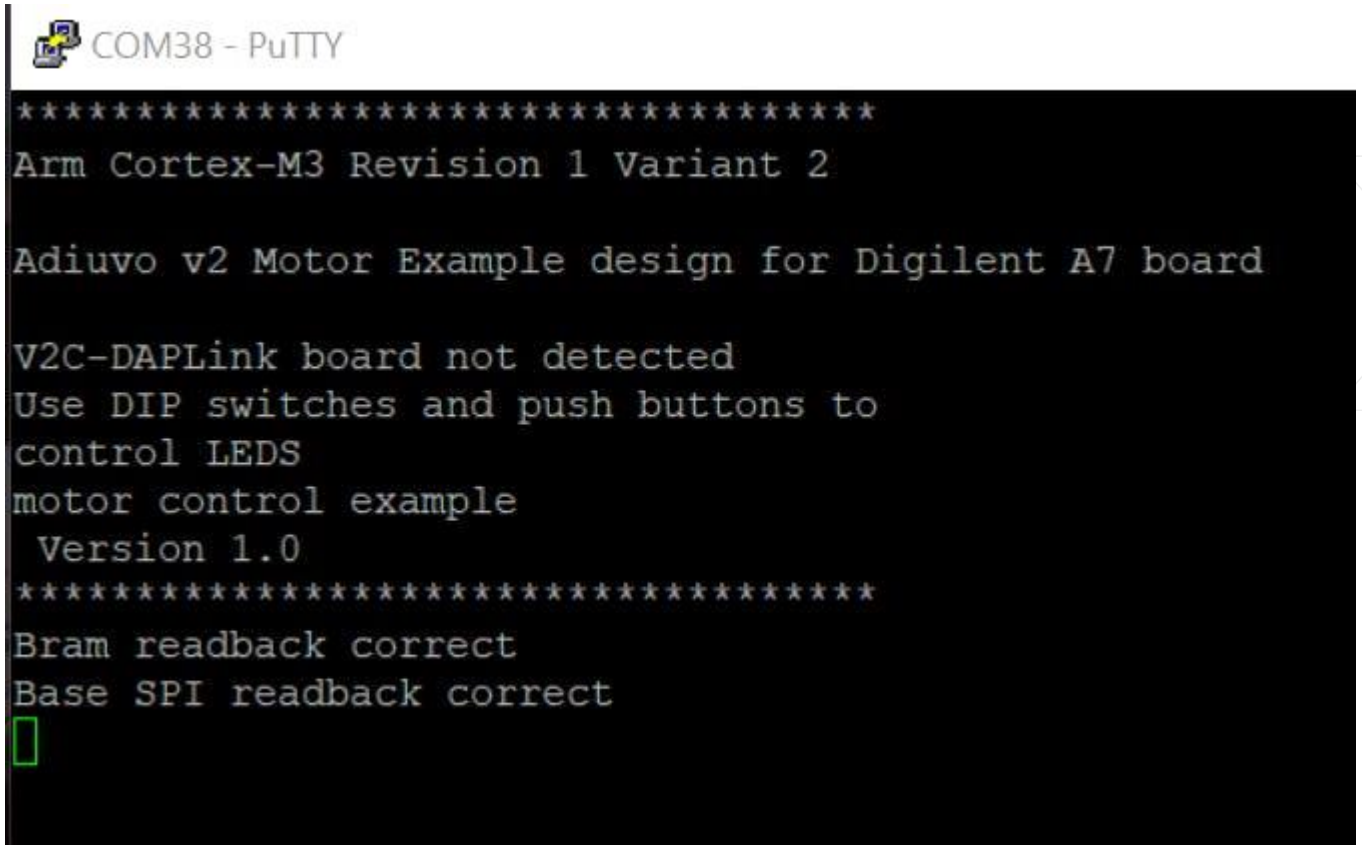
# Programming the Arty

- Use Vivado Hardware Manager to program the Arty



# Programming the Arty

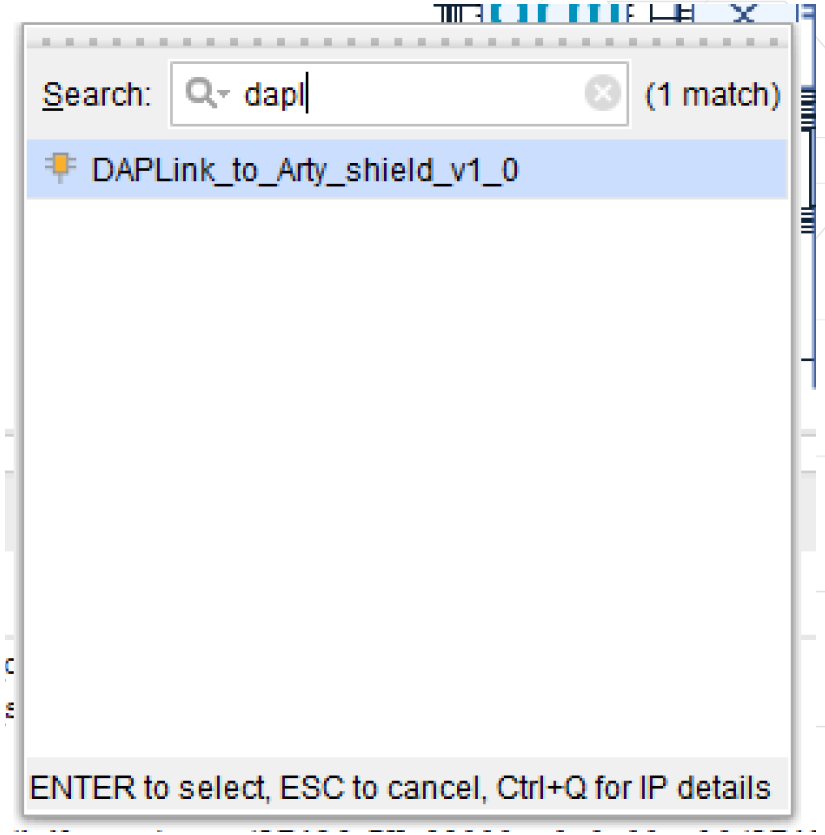
- **Once downloaded the program will begin executing immediately, if you have terminal connected you will see your terminal output appear**



# Lab Two
# Saying Hello World (20 Minutes)

# Debugging

- **We can debug the ARM M1 and M3 cores using either**
  - Serial Wire and the DAP Link module on the Arty Boards
  - Standard alone JTAG / Serial Wire
- **Connections made in Vivado Design**
  - JTAG just breaking out pins to IO
  - DAP Link more complex
- **DAP Link IP Core is provided**

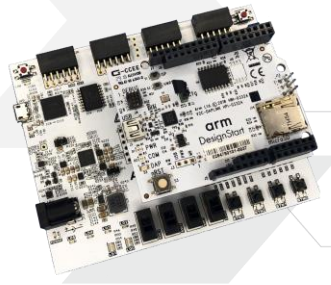


# Debug on Arm V2C-DAPLink board

- **Use Arm® DesignStart DAPLink board for Serial Wire Debug (SWD) and UART access to any design built on this board**

- **Debug Tips**
  - Serial Wired Debug (SWD) through DAPLink USB connector
  - Drag and Drop programming interface for quick *.bin file updates
  - Use Additional QSPI and SD port interface options
    - Dedicated QSPI flash can be used for code image
    - Micro-SD card for application use (SPI mode only)





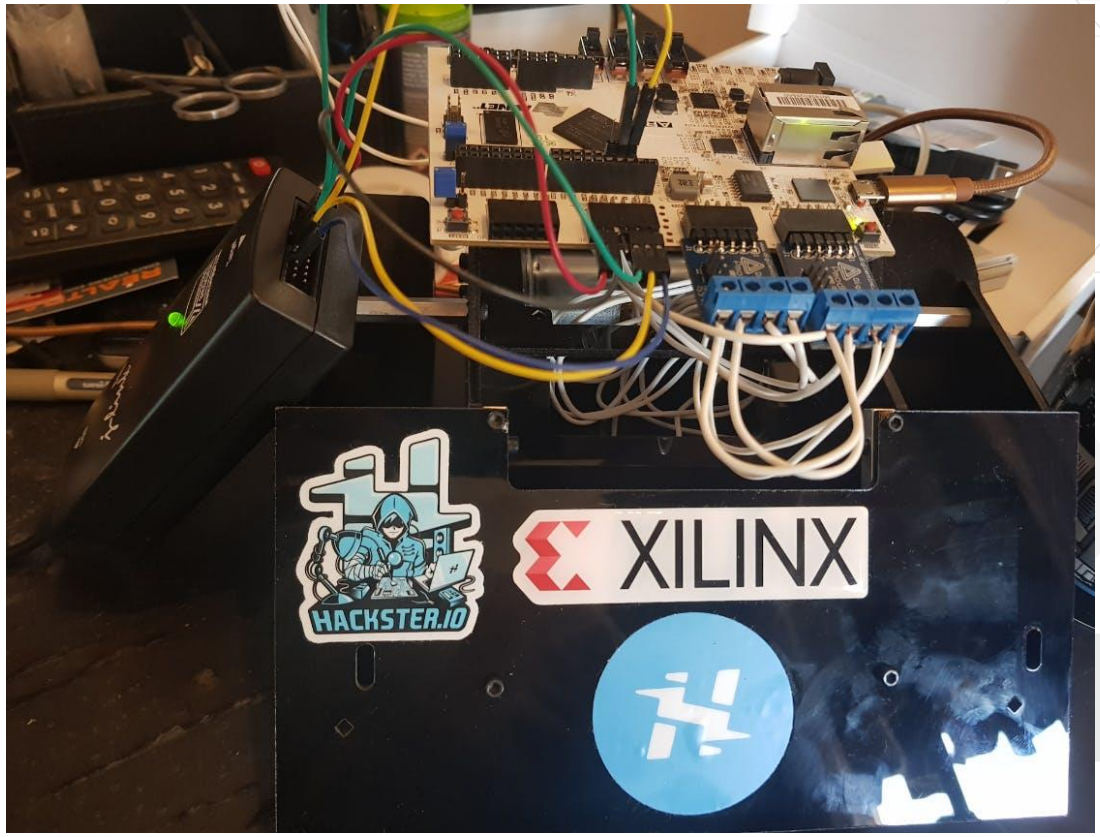
# DAP Link In Memory

Address Editor | impl_1_place_report_io_0 - impl_1

| Cell | Slave Interface | Base Name | Offset Ad... | Range | High Address |
|---|---|---|---|---|---|
| Cortex_M3_0 | | | | | |
| CM3_SYS_AXI3 (32 address bits : 4G) | | | | | |
| daplink_if_0/axi_xip_quad_spi_0 | AXI_LITE | Reg | 0x4000_0000 | 64K | 0x4000_FFFF |
| daplink_if_0/axi_gpio_0 | S_AXI | Reg | 0x4001_0000 | 64K | 0x4001_FFFF |
| daplink_if_0/axi_quad_spi_0 | AXI_LITE | Reg | 0x4002_0000 | 64K | 0x4002_FFFF |
| daplink_if_0/axi_single_spi_0 | AXI_LITE | Reg | 0x4003_0000 | 64K | 0x4003_FFFF |
| axi_uartlite_0 | S_AXI | Reg | 0x4010_0000 | 64K | 0x4010_FFFF |
| axi_gpio_0 | S_AXI | Reg | 0x4011_0000 | 64K | 0x4011_FFFF |
| axi_gpio_1 | S_AXI | Reg | 0x4012_0000 | 64K | 0x4012_FFFF |
| axi_quad_spi_0 | AXI_LITE | Reg | 0x4013_0000 | 64K | 0x4013_FFFF |
| PmodGPIO_0 | AXI_LITE_GPIO | Reg0 | 0x4014_0000 | 4K | 0x4014_0FFF |
| PmodGPIO_1 | AXI_LITE_GPIO | Reg0 | 0x4015_0000 | 4K | 0x4015_0FFF |
| PmodMAXSONAR_0 | AXI_LITE_GPIO | Reg0 | 0x4016_0000 | 4K | 0x4016_0FFF |
| axi_bram_ctrl_0 | S_AXI | Mem0 | 0x6000_0000 | 8K | 0x6000_1FFF |
| CM3_CODE_AXI3 (32 address bits : 4G) | | | | | |
| daplink_if_0/axi_xip_quad_spi_0 | AXI_FULL | MEM0 | 0x0000_0000 | 1M | 0x000F_FFFF |

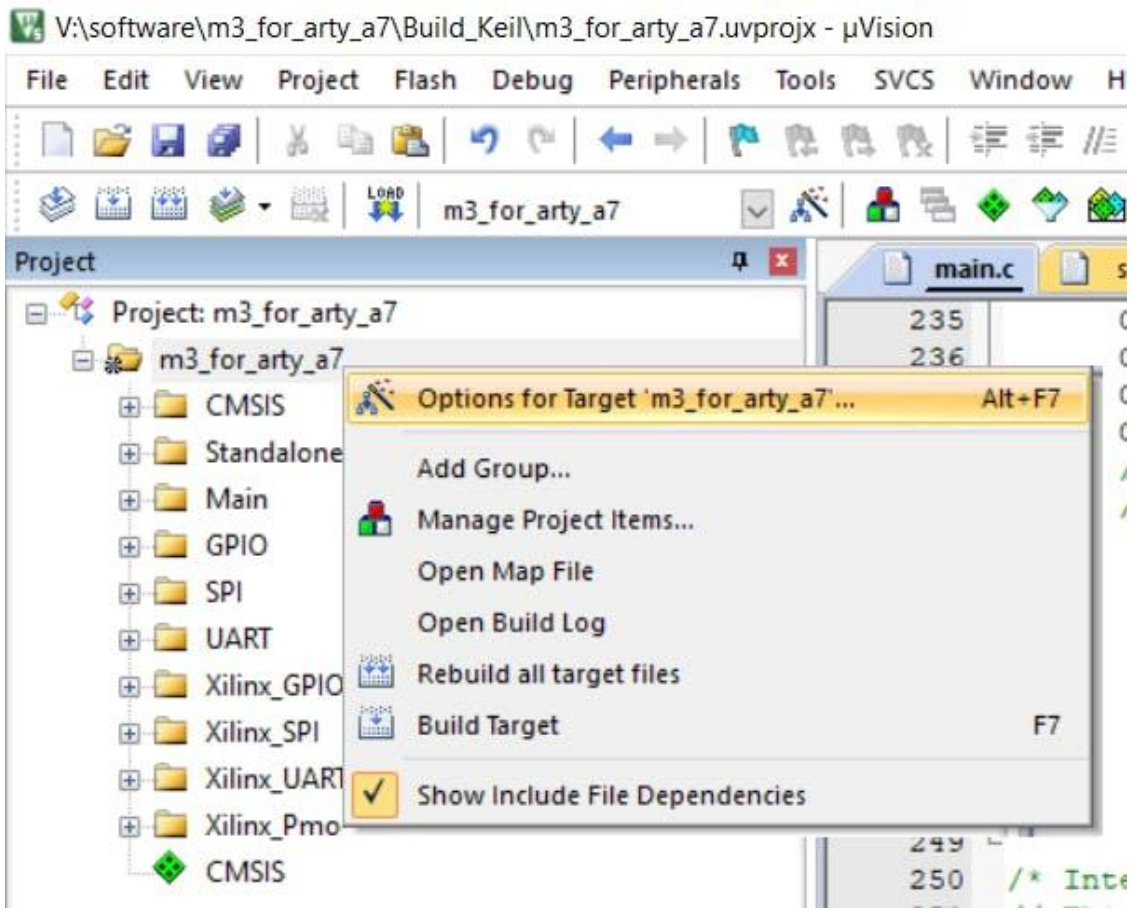
# Debug Using JTAG

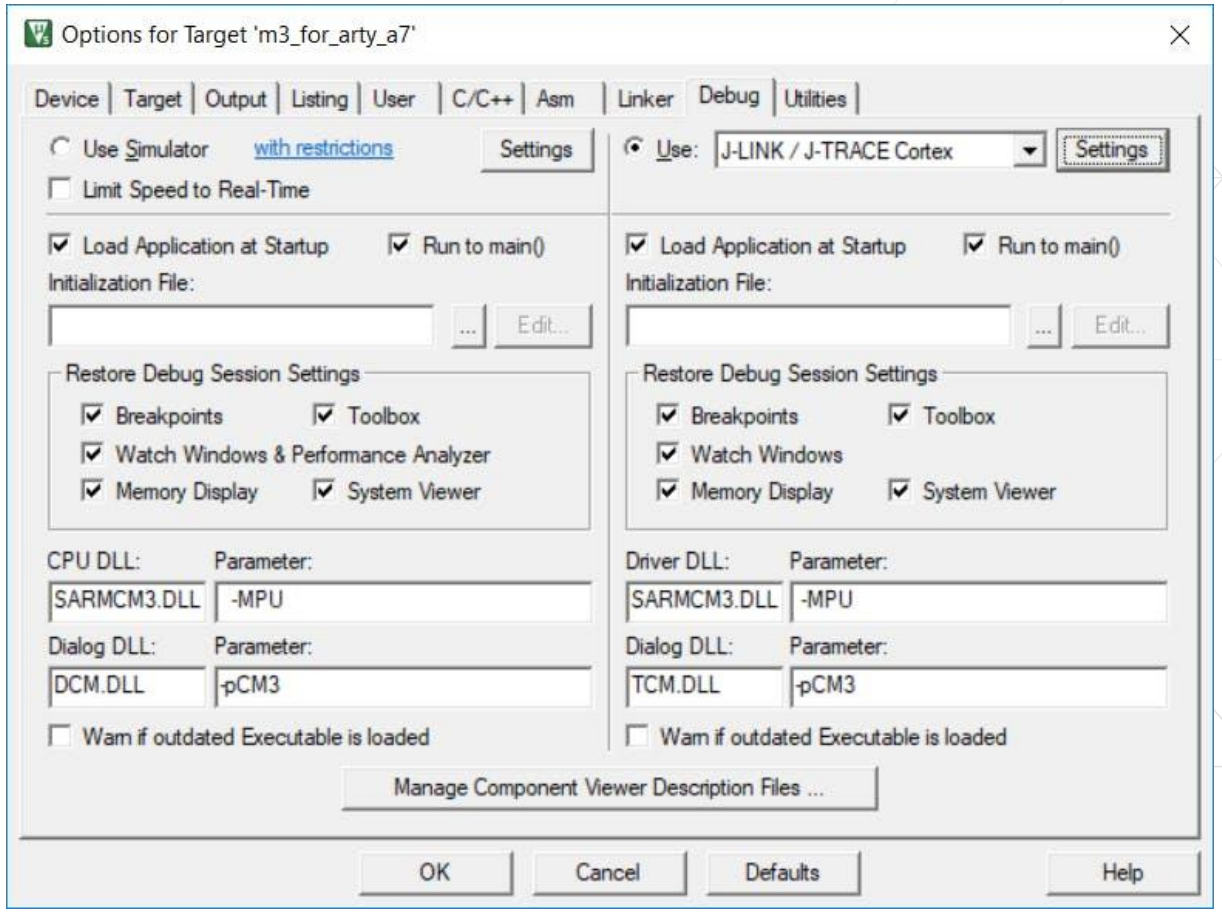
- JTAG pins are broken out on the Arty board to both PmodC and the Shield Connector
  - TMS - Shield IO-40
  - TCL - Shield IO-41
  - nTRST - PmodC Pin 1
  - TDI - PmodC Pin 2
  - TDO - PmodC Pin 3



# Configuring Debug

> **Use options to define the debug mode**



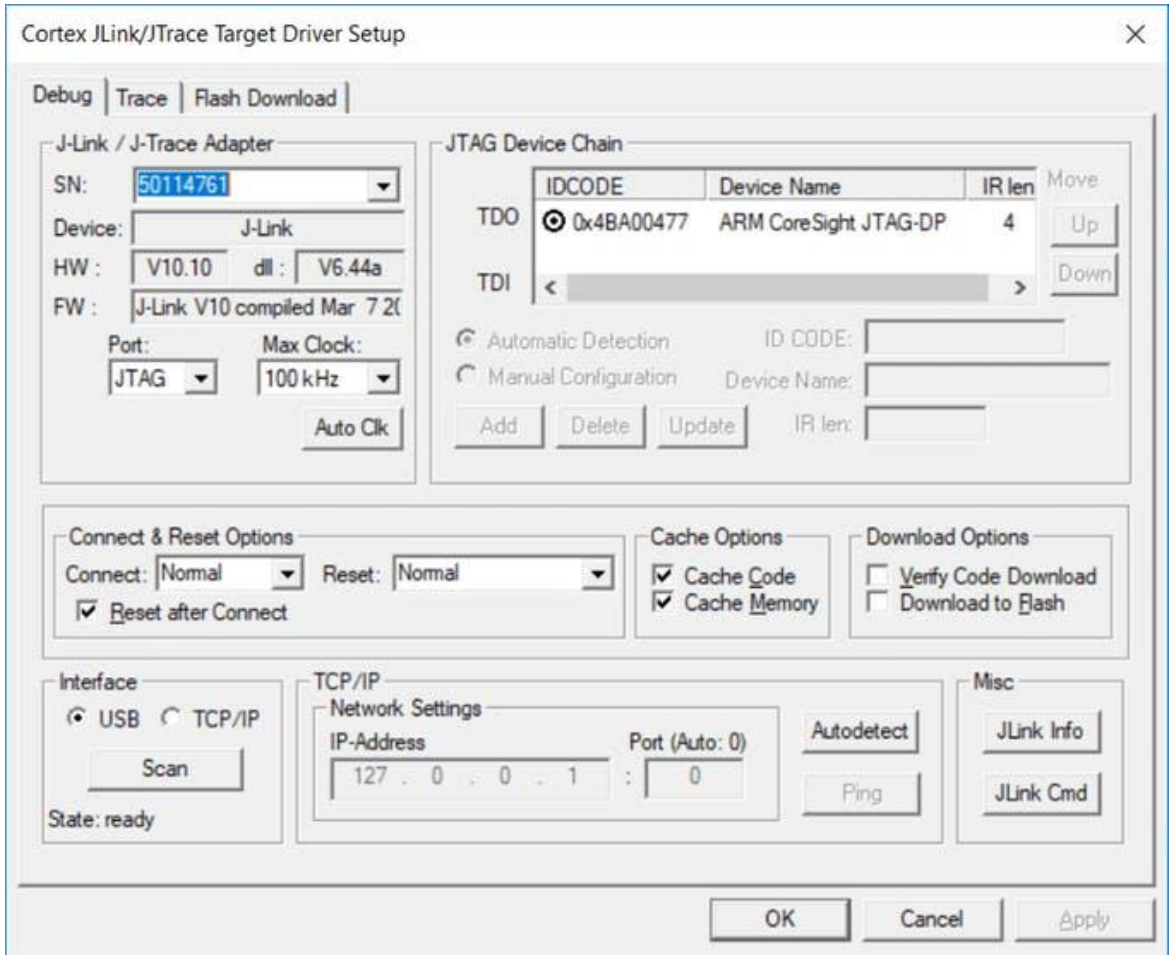





XILINX.

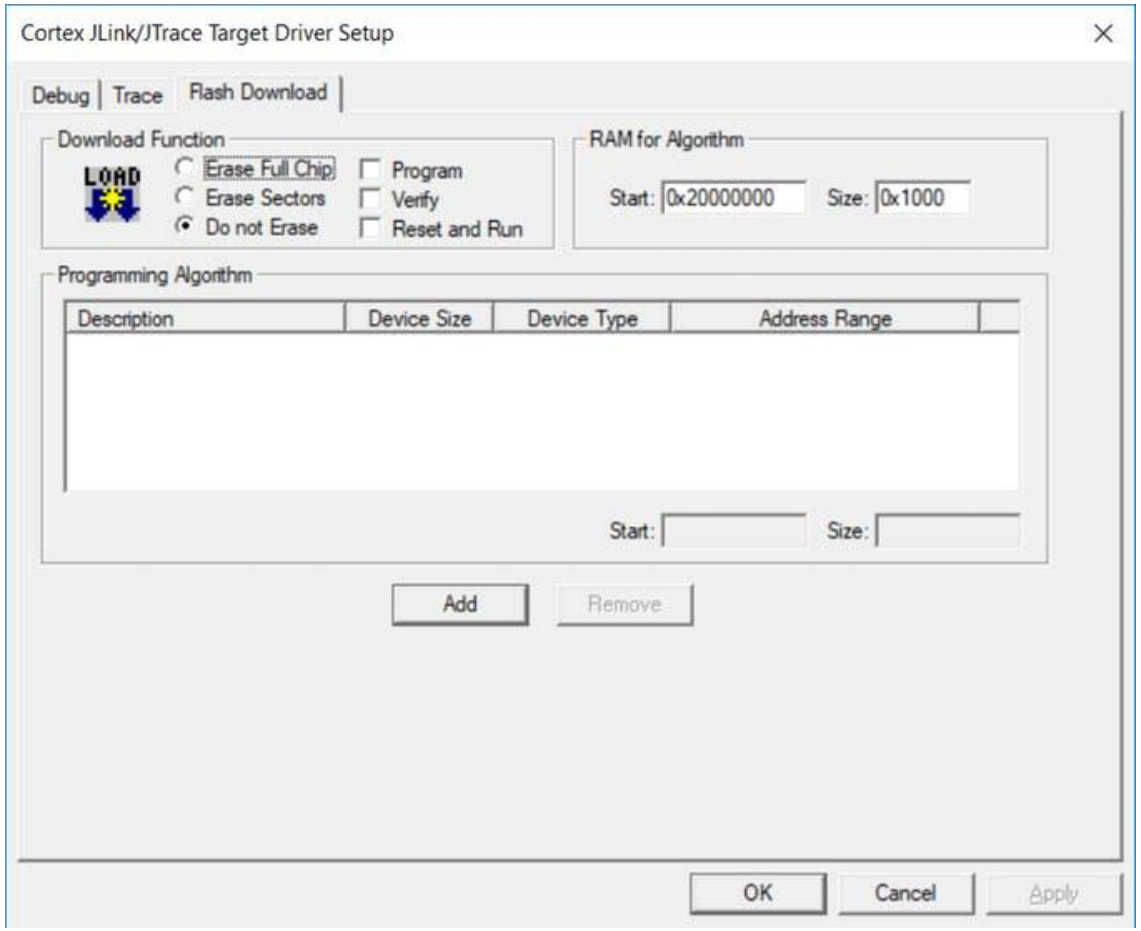
# Debug Setting

> To configure the settings on your chosen debug tool click settings

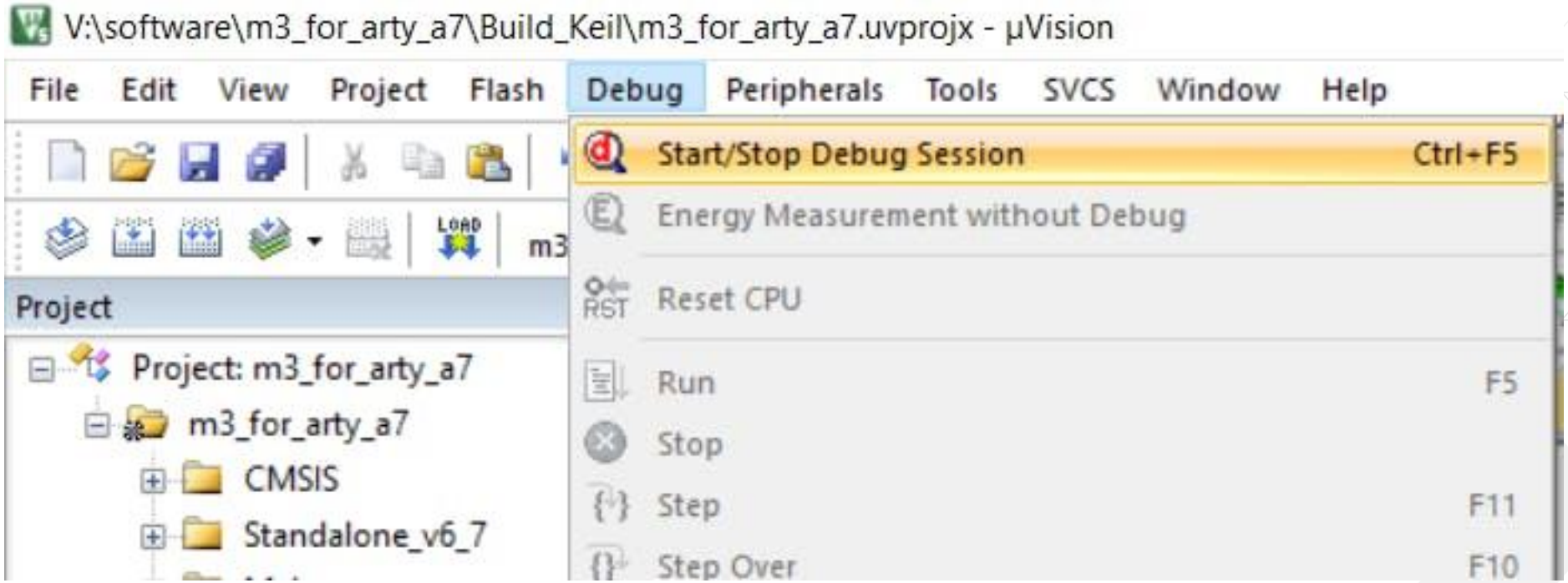


# Flash Settings

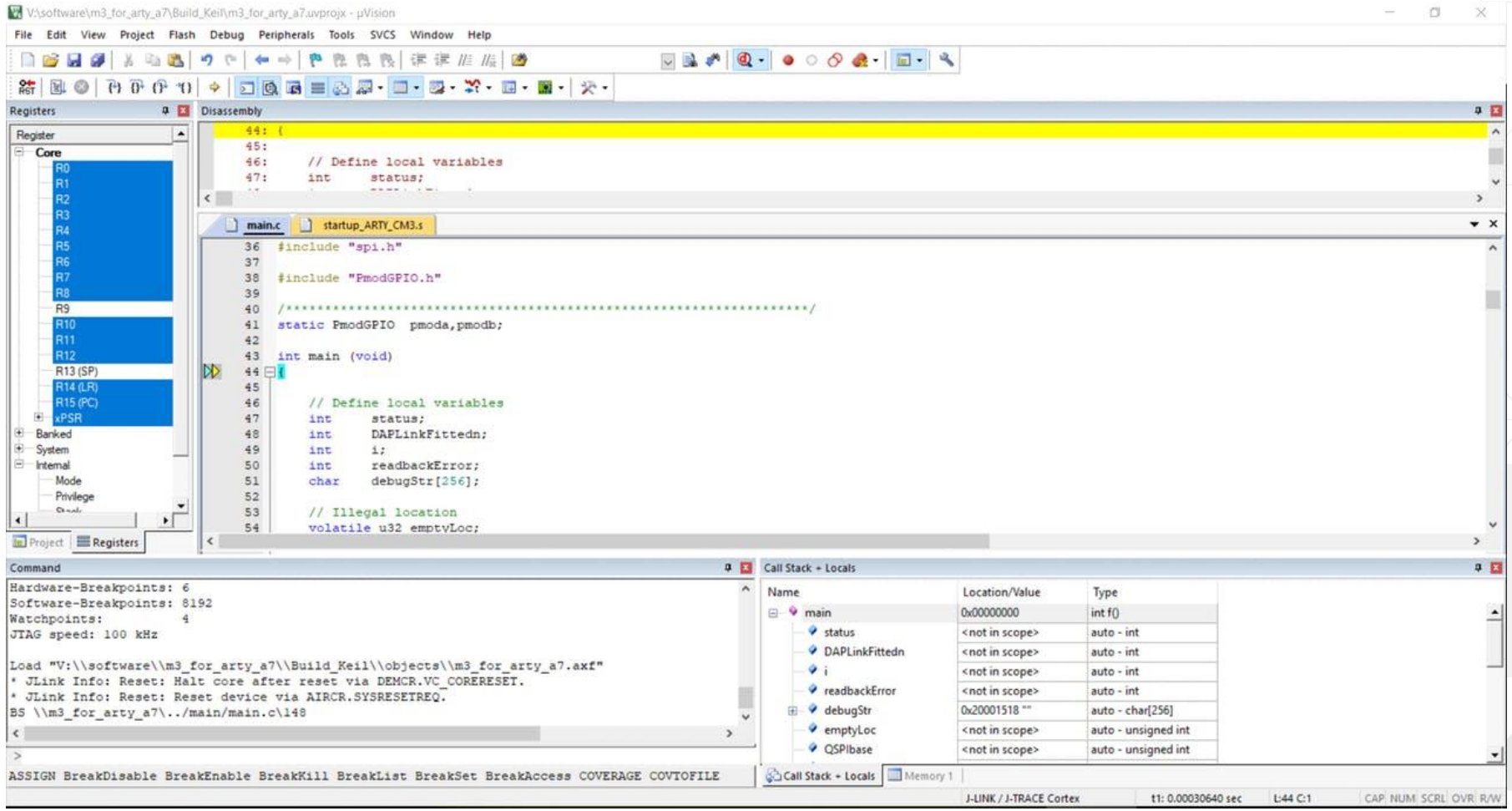
> **If desired we can download to flash, the application.**



# Starting & Stopping Debug



# Debug Environment



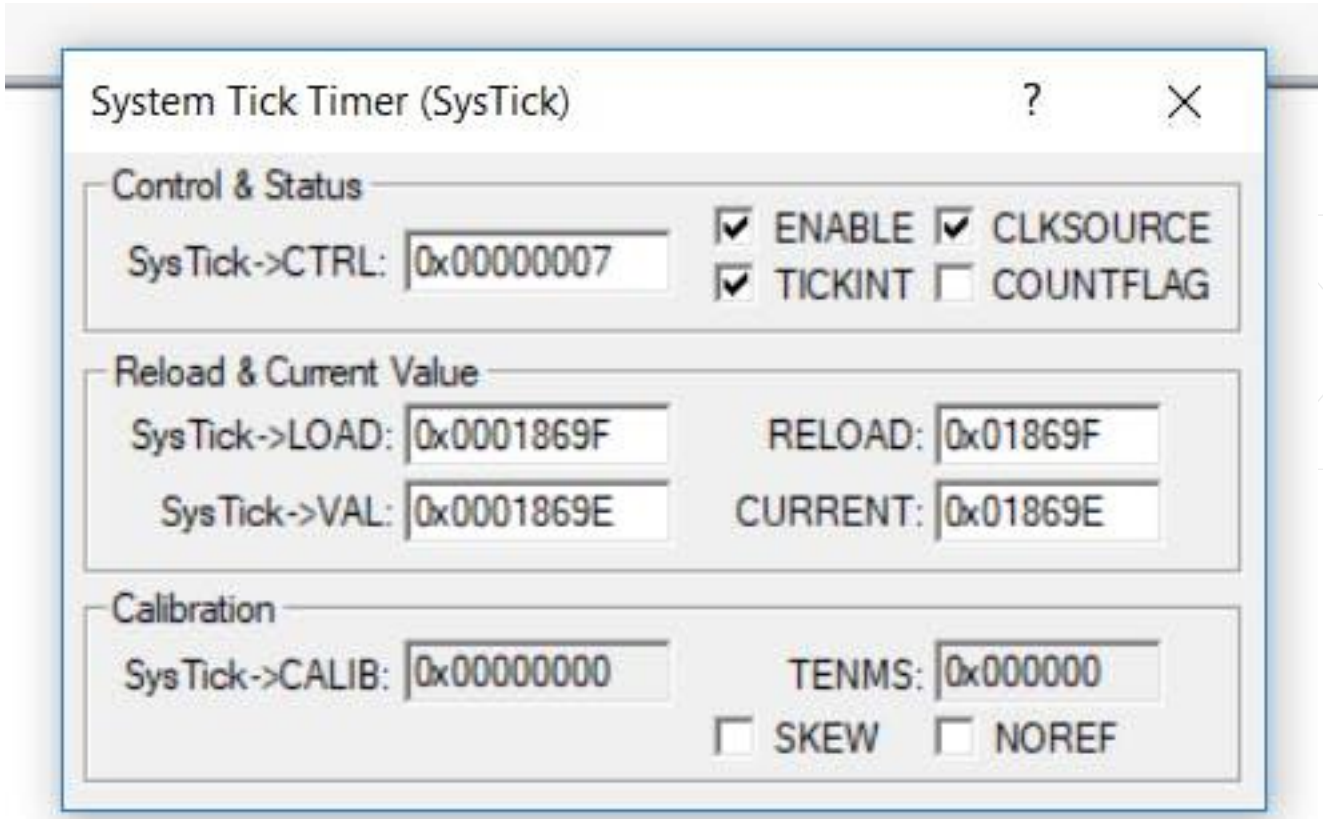





XILINX.

# Monitoring Registers Arm Kiel

> **Once the debugger is set up we can measure system registers e.g. System Tick**



# Coffee Break (15 Minutes)

# Detailed Lab

- **Objective of the detailed Lab is to work with Vivado, SDK, Arm Keil to create a simple environmental monitoring system.**

- **We will use the Arty S7, Arm Cortex M1 and the PmodNAV & PmodHYGRO**

- **PmodNAV and PmodHYGRO enable you to obtain**
  - Temperature
  - Humidity
  - X,Y,Z Acceleration
  - Pressure / Altitude
  - Compass heading
  - X,Y,Z Rotation

- **Full instructions are provided in the lab sheet – ASK if you need help**

# Lab Three (80 Minutes)

# Adaptable.
# Intelligent.

XILINX