System on Chip

A processor based System-on-Chip will typically include:
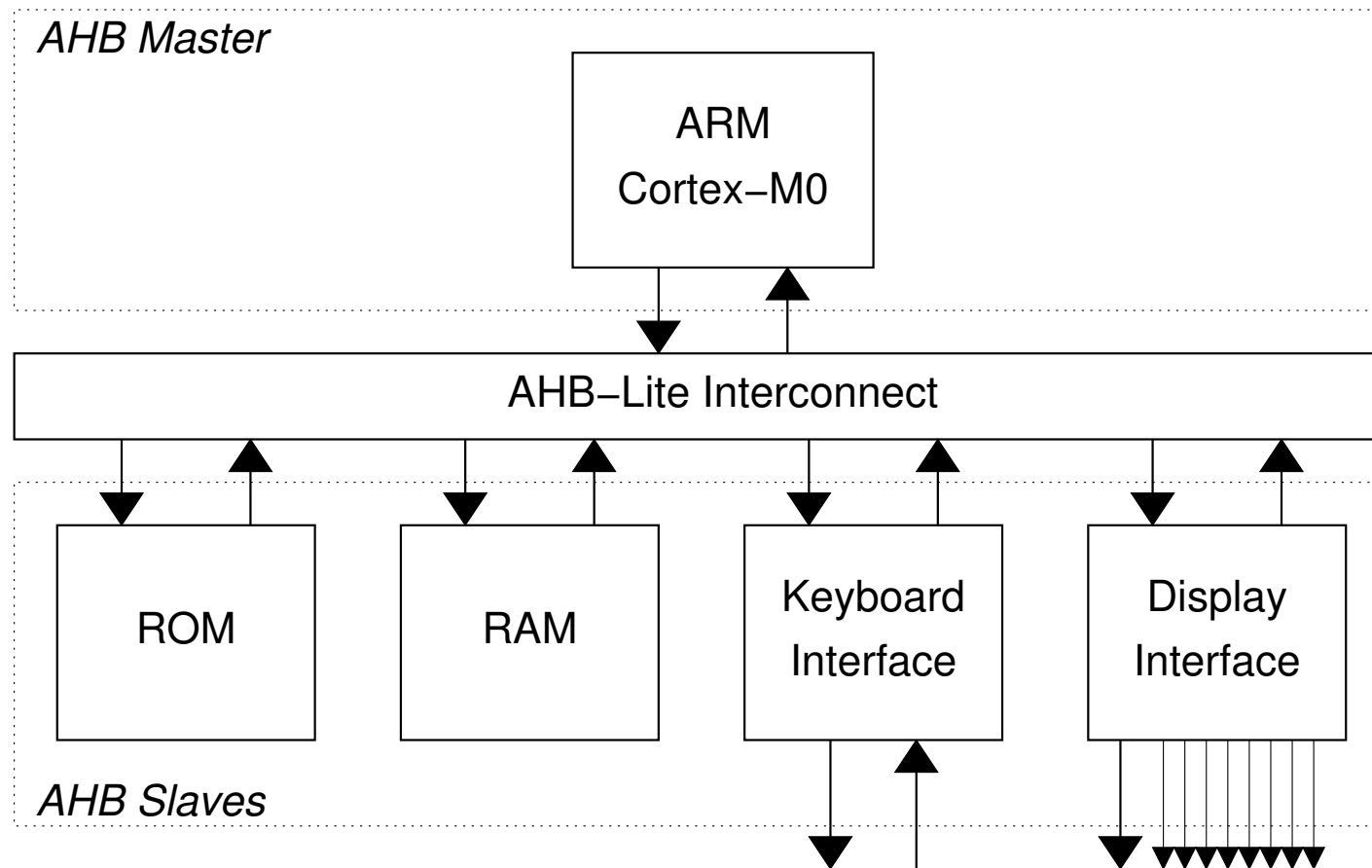
- Hardware

  - A processor core (e.g. ARM M0 DesignStart, RISC-V PicoRV32)
  - Interconnect (e.g. AHB-Lite bus)
  - Fixed program memory (ROM)
  - Data memory (RAM)
  - Application specific interfaces (for input and ouput)

- Software
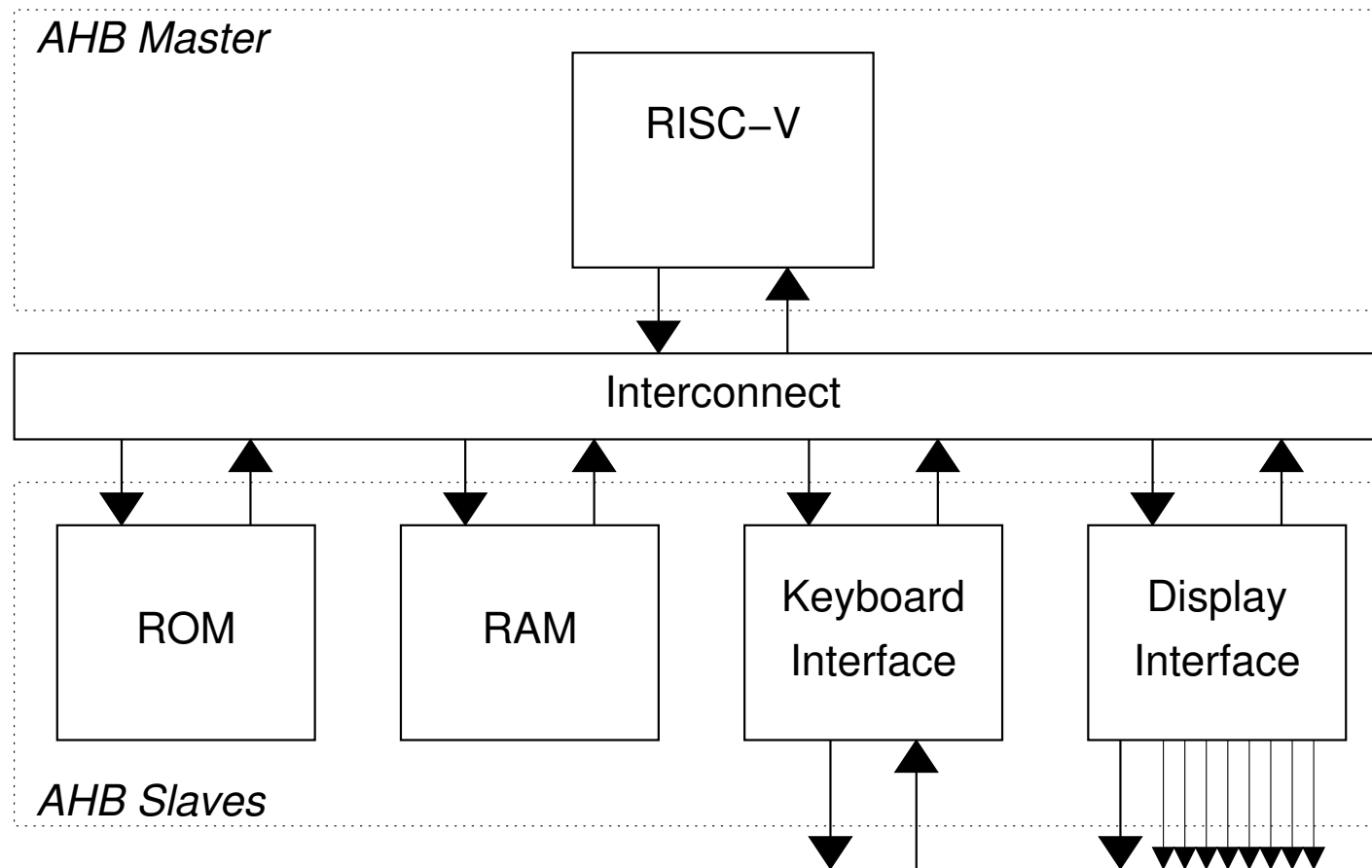
  - Application specific software (typically written in C)
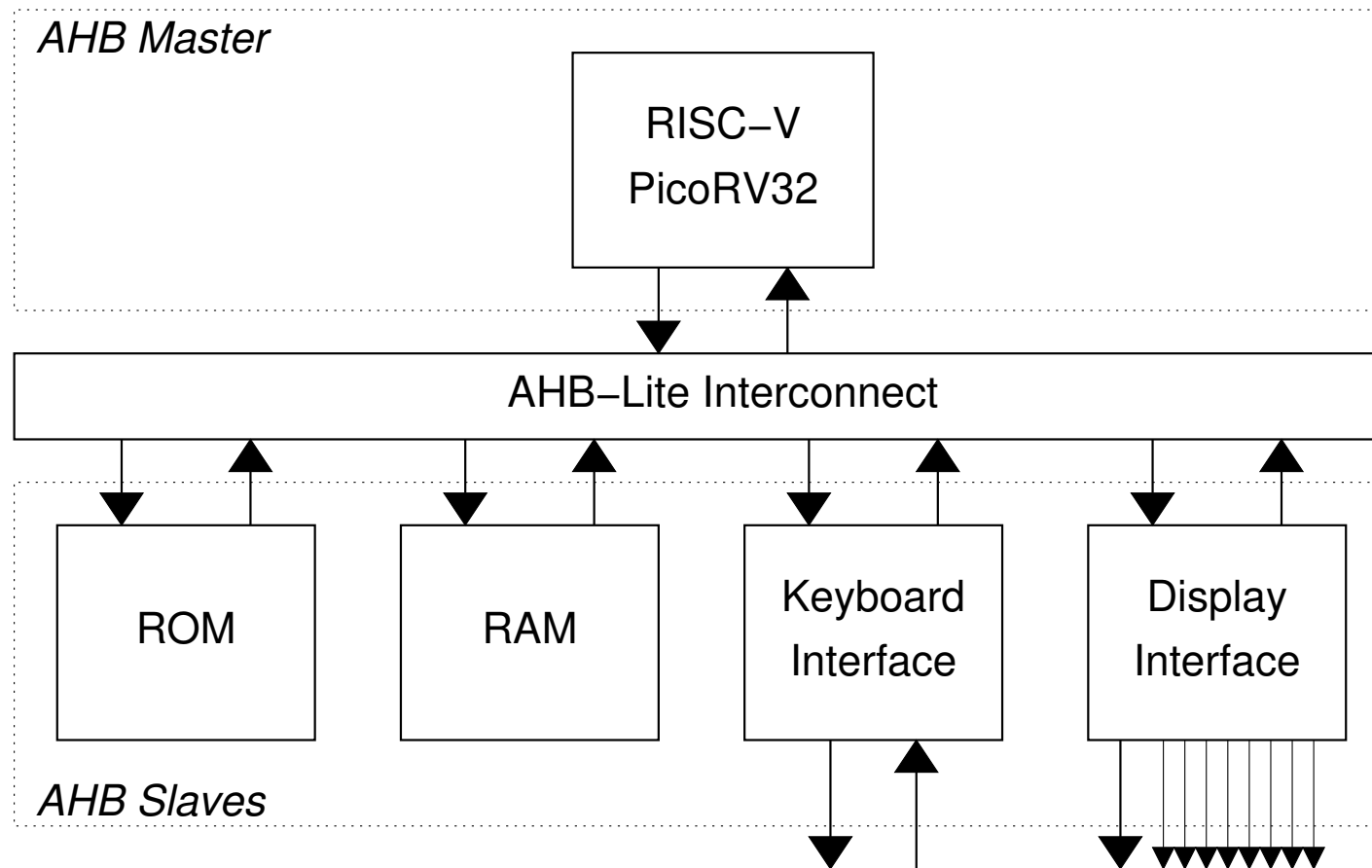
System on Chip

Hardware

AHB Master

ARM
Cortex–M0

AHB–Lite Interconnect

ROM RAM Keyboard Interface Display Interface

AHB Slaves

System on Chip

Hardware

System on Chip

Hardware

System on Chip

Use of Third-Party Intellectual Property

The use of hardware and software modules created by others allows designers to build larger and more complex systems.
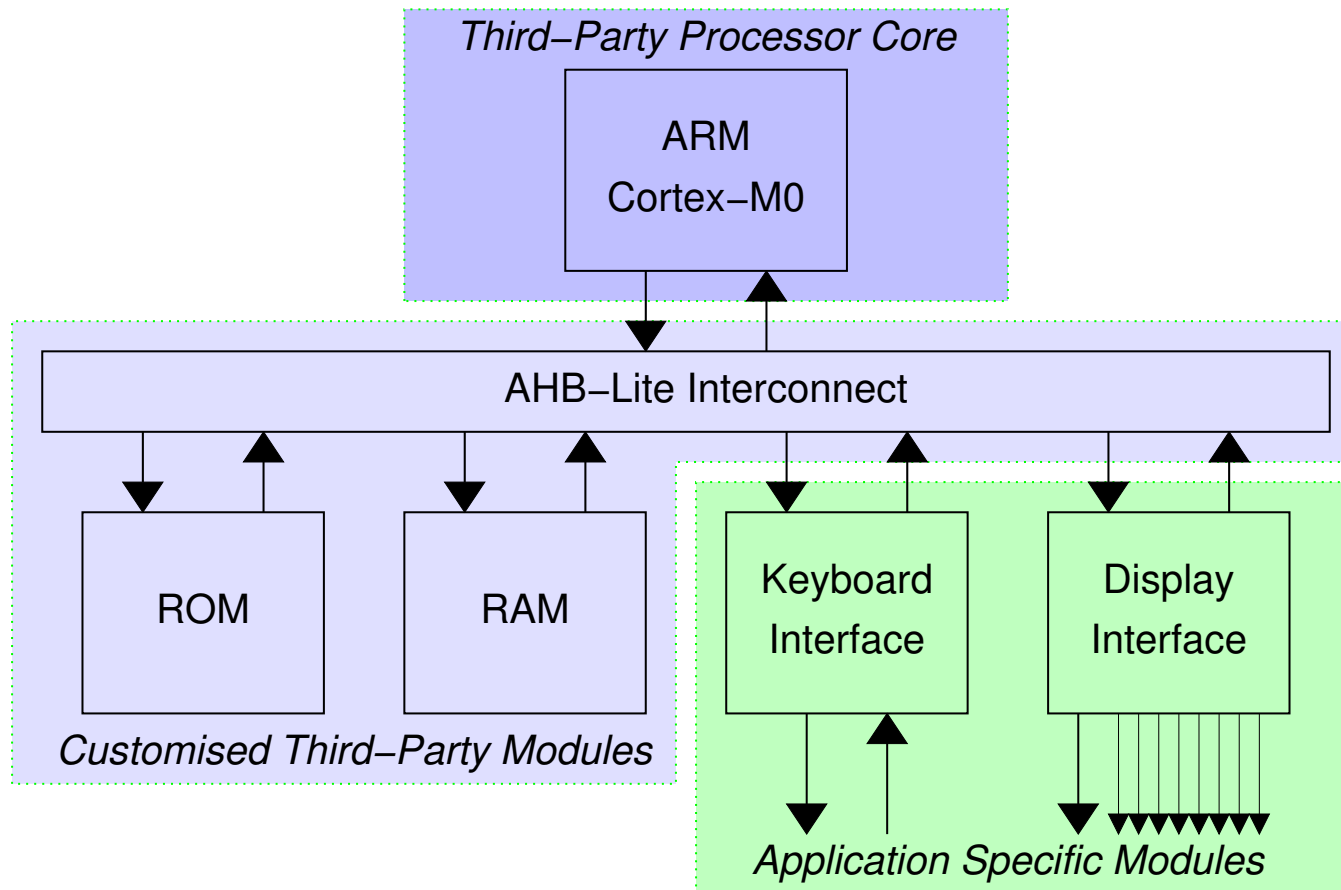
You may make use of third-party intellectual property in your design provided that:

1. You have the permission of the designer
2. You acknowledge its use in your report
3. You do not remove copyright/licensing notices

In a simple design you might expect to use a third-party processor core together with customised versions of third-party interconnect, ROM and RAM modules

System on Chip

Use of Third-Party Intellectual Property

Third–Party Processor Core

ARM

Cortex–M0

AHB–Lite Interconnect

ROM

RAM

Keyboard Interface

Display Interface

Customised Third–Party Modules

Application Specific Modules
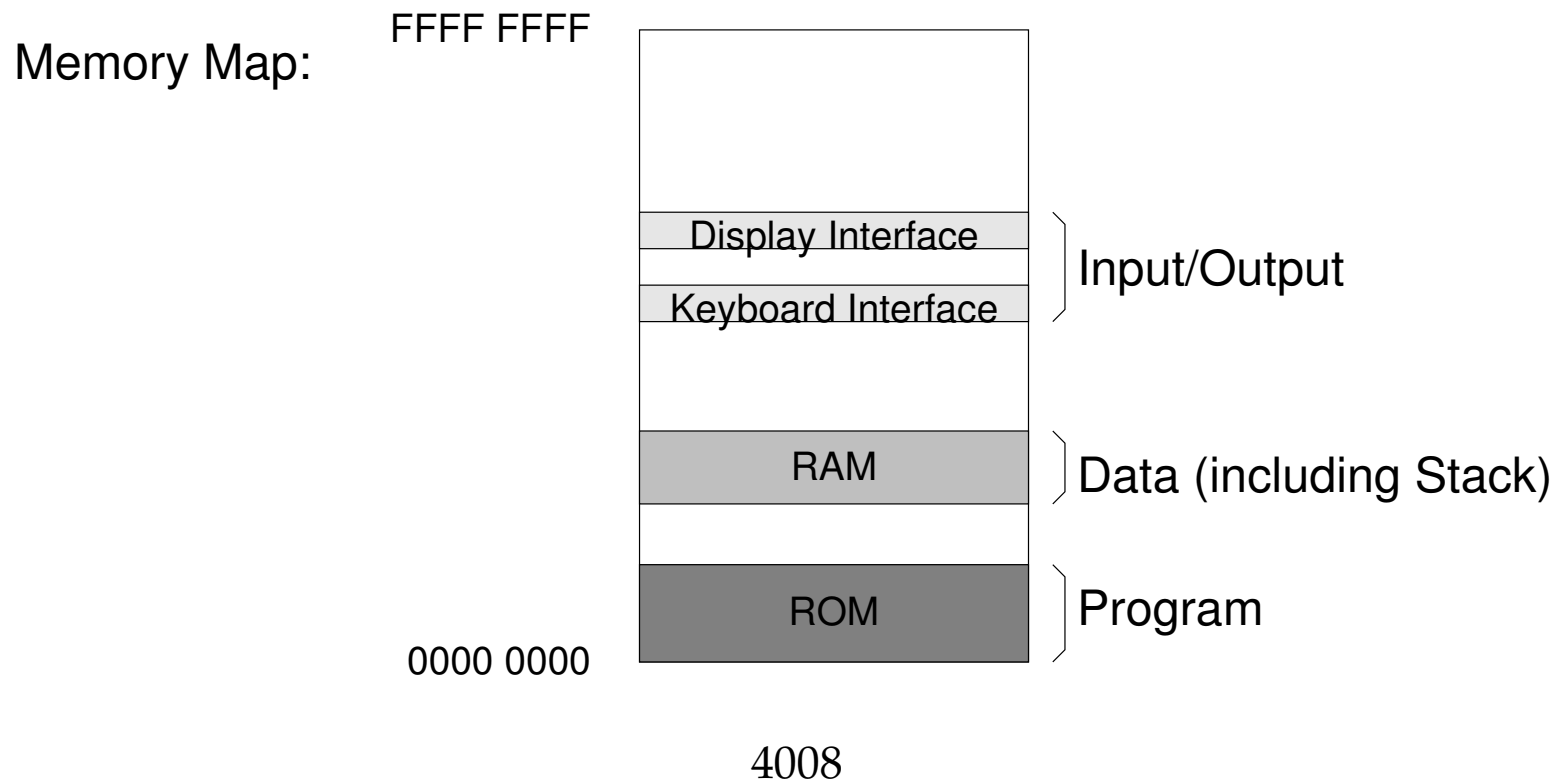
System on Chip

ARM Cortex-M0 DesignStart

- Simple 32-bit ARM processor core

  Designed for embedded applications.

- Executes 16-bit instructions from the ARM Thumb2 instruction set

  Supported by standard GNU compilers.

- Obfuscated Verilog Source Code

  Designed to be usable but not to be reverse engineered or customised.

- Available free for academic and other non-commercial use

  You may use the provided core for this project. If you wish the Cortex-M0 DesignStart for another purpose, you should apply to ARM for permission.

System on Chip

Programmer's View - Memory Mapped I/O

The programmer sees each input/output device as occupying one or more memory locations.

Memory Map:

FFFF FFFF

Display Interface	
Keyboard Interface	Input/Output
RAM	Data (including Stack)
ROM	Program

0000 0000

4008

System on Chip

AHB-Lite Bus

Each memory access takes two or more clock cycles

HCLK

HTRANS[1]

HADDR   Address A   Address B

HWRITE

HWDATA   Data B

HREADY

HRDATA   Data A

2–Cycle Memory
Access (Read)

Memory Access (Write)
Extended to 3 Cycles
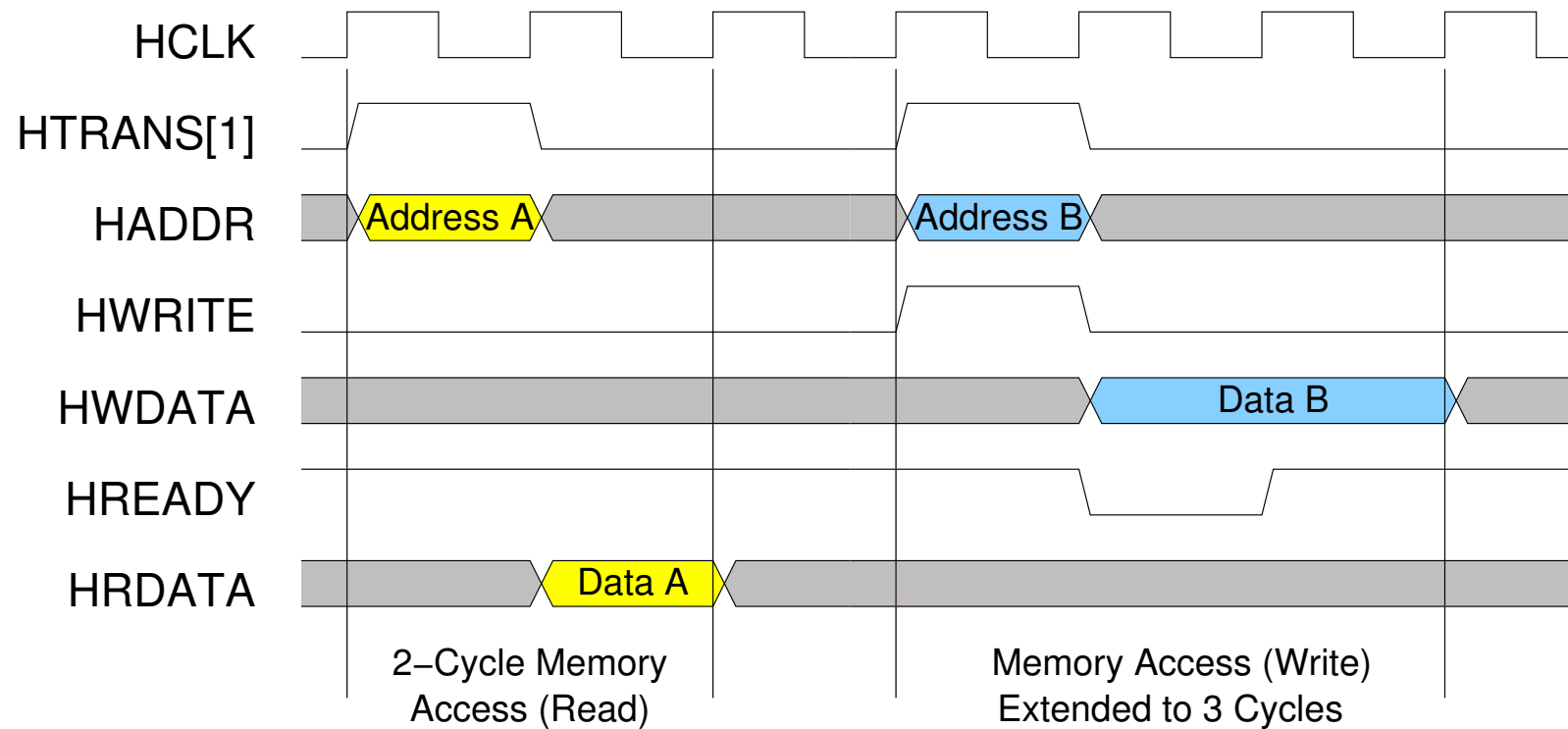
System on Chip

AHB-Lite Bus

Memory accesses can be pipelined[1]

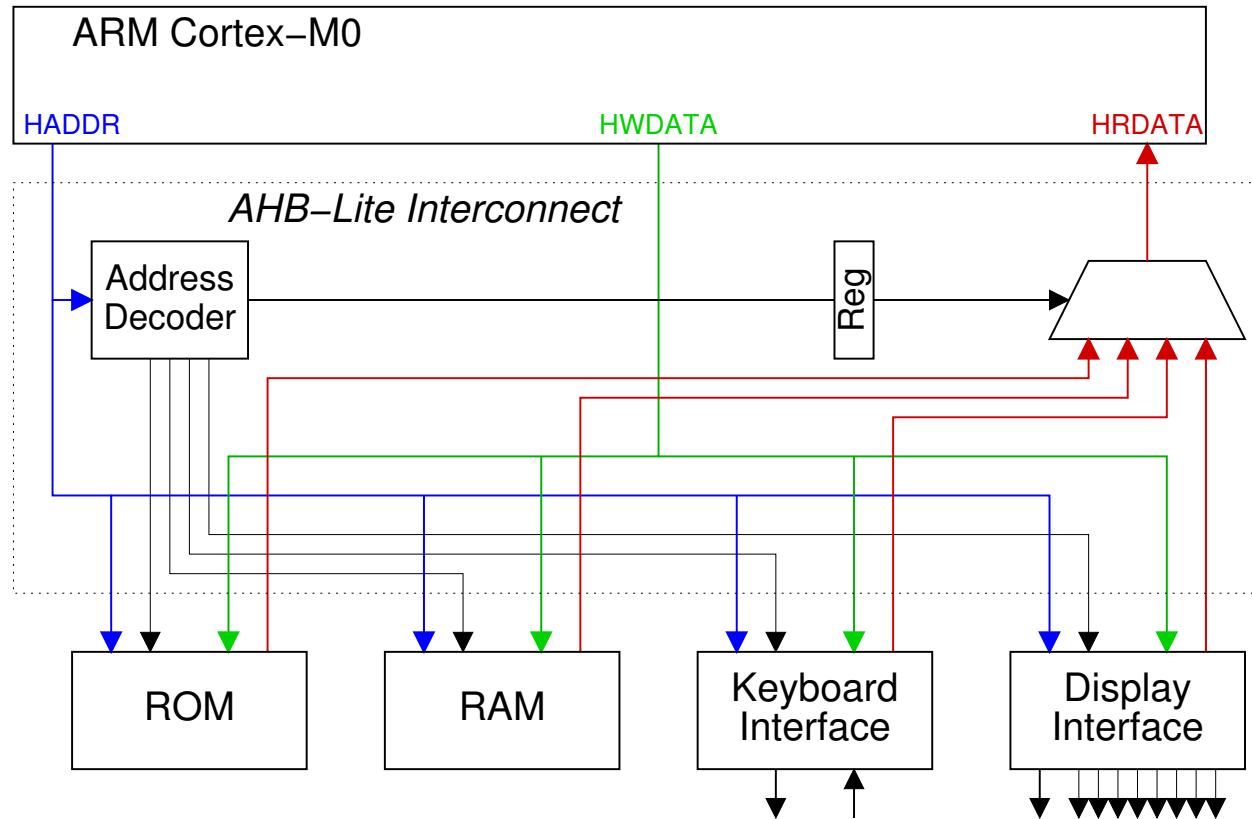| HCLK	
HTRANS[1]	
HADDR	Address A, Address B, Address C, Address D, Addr..
HWRITE	
HWDATA	Data B, Data C, Data D
HREADY	
HRDATA	Data A
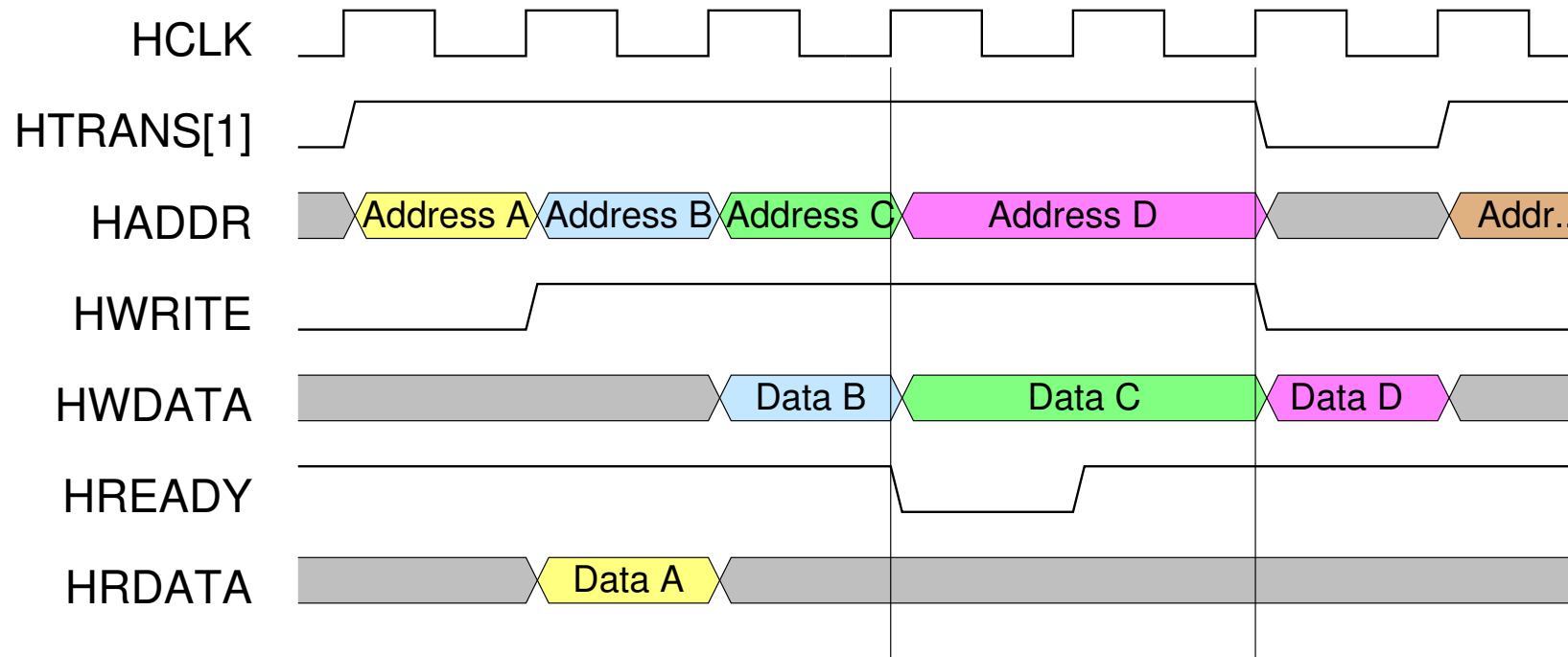
* The wait state for transaction C also extends the address phase of transaction D

---

[1]the address phase of one memory access may start before the data phase of the previous access is complete

System on Chip

AHB-Lite Interconnect

showing HADDR, HWDATA, HRDATA and individual slave HSEL select signals.

4011

System on Chip

Memory Map:

FFFF FFFF

A[31:30] == 3

Display Interface

A[31:30] == 2

Keyboard Interface

A[31:30] == 1

RAM

A[31:30] == 0

ROM

0000 0000
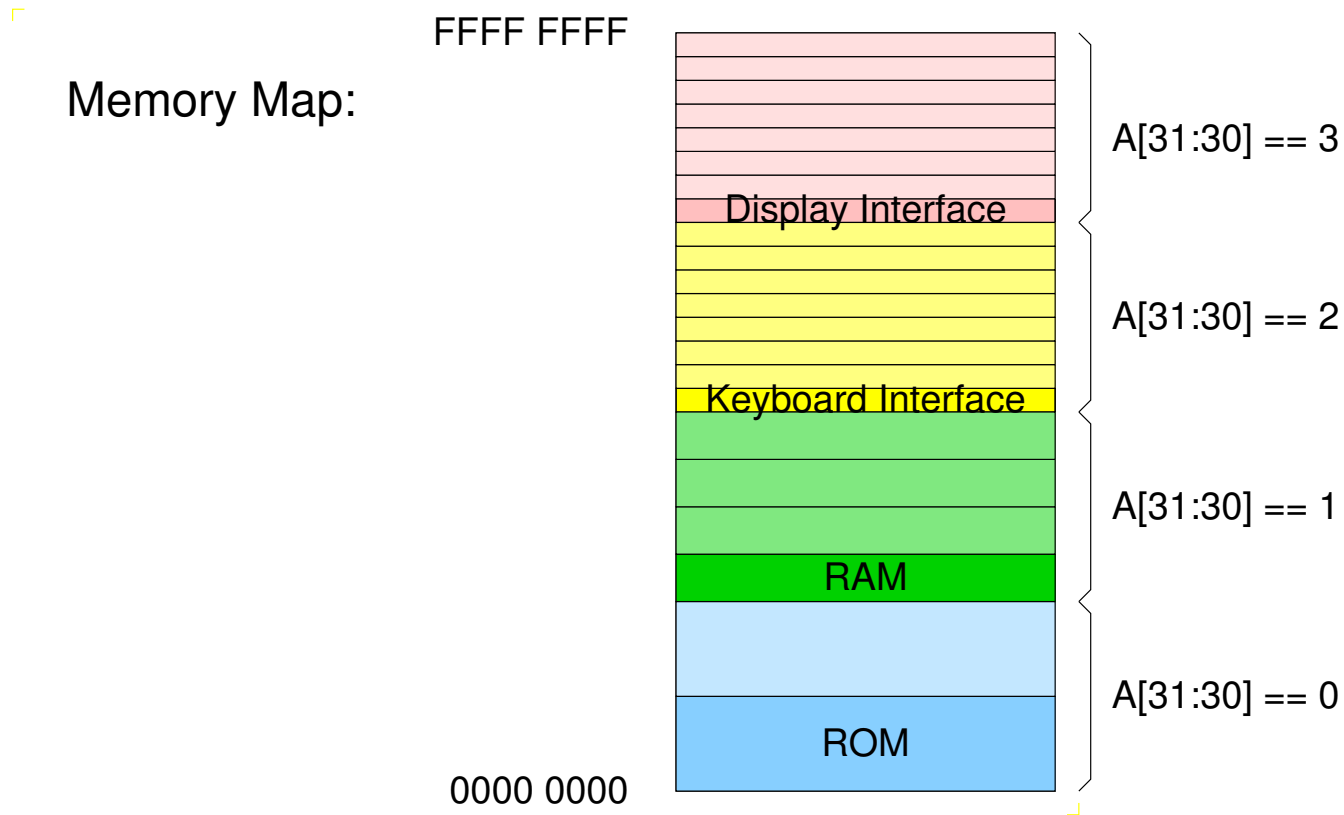
- With partial address decoding we use fewer address lines (A31 and A30 in the example above) and less logic for decoding.

- A side effect of this is that we get multiple images of some or all of the slave devices.

Application Specific Interface Module Design

- Decide on Module Function

  - Some functions are easy in hardware others are easy in software

    While non-specific input and output ports can be used (in the style of a microcontroller i/o ports), this is not usually a good use of system-on-chip resources.

- Decide on the Programmer's Model

  - What Registers?
    Input registers, Output registers, Status registers
  - What Addresses?
  - What side effects?
    e.g. Access to input or output register $\Rightarrow$ changes status

System on Chip

Interface Module Registers

AHB−Lite Interconnect

PS2 Keyboard Interface

Address Offset		Read/Write
+0	Status	ro
+4	KeyData	ro

VGA Display Interface

Address Offset		Read/Write
+0	Pixel_row_0_col_0	rw
+4	Pixel_row_0_col_1	rw
+8	Pixel_row_0_col_2	rw
	640x480 *Frame Buffer*	
+1,228,788	Pixel_row_479_col_637	rw
+1,228,792	Pixel_row_479_col_638	rw
+1,228,796	Pixel_row_479_col_639	rw

Serial Link to PS2 Keyboard

Cable to VGA Monitor