C3

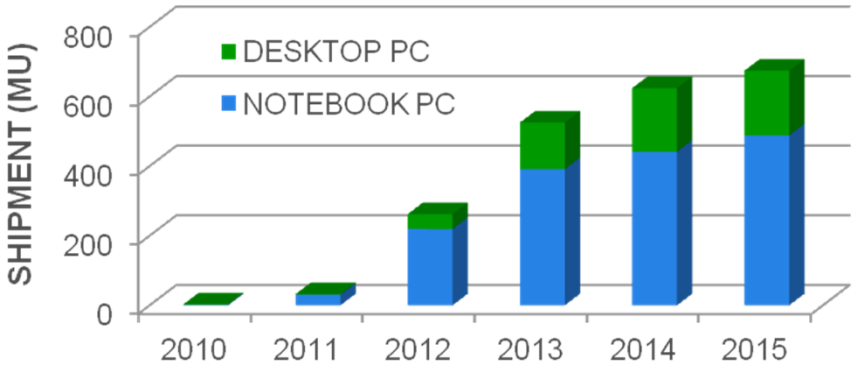
Cypress Semiconductor USB3.0 Solution

Ken Higashiyama

June 13, 2012

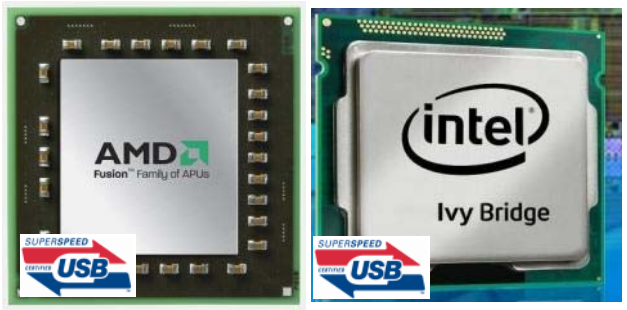
USB 3.0 Ecosystem

	2012	2013	2014	2015
USB 3.0 PC SHIPMENT	262 MU	525 MU	624 MU	673 MU
USB 3.0 PENETRATION	50.3%	90.7%	99.6%	100%



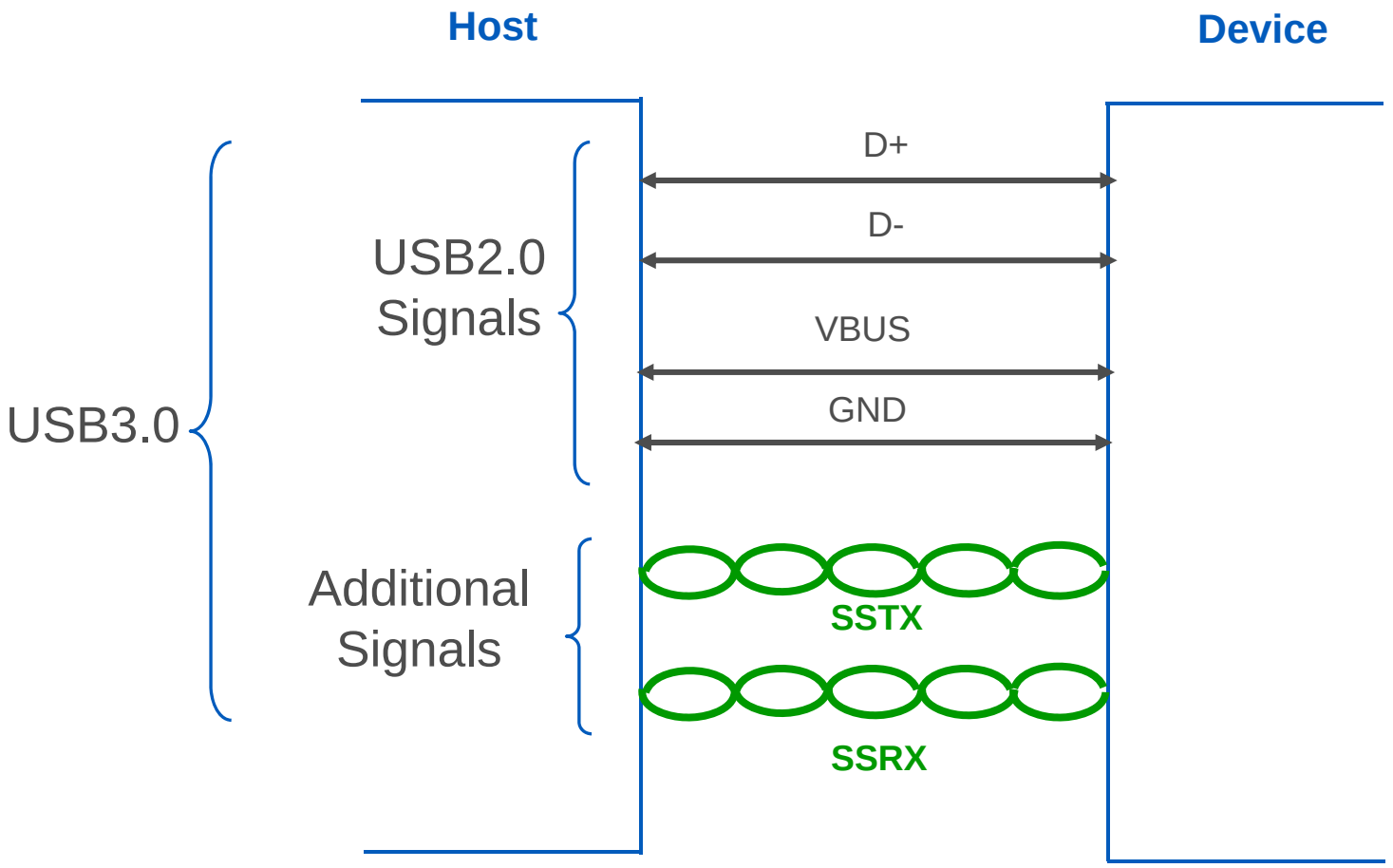
Source: In-Stat Wired USB May, 2011

- AMD SHIPPED ITS 1ST USB 3.0 CHIPSET (A75, A70M) APRIL 2011
- INTEL LAUNCHED ITS IVY BRIDGE CPU AND COMPANION USB 3.0 CHIPSET (Z77) APRIL 2012



446 USB 3.0 SUPERSPEED PRODUCTS CERTIFIED TO DATE

USB 3.0 Interface

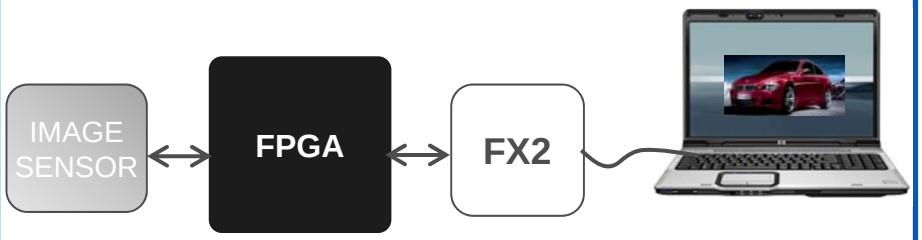


USB 3.0 has two pairs of differential signals in addition to USB 2.0 signals

USB 3.0 Applications

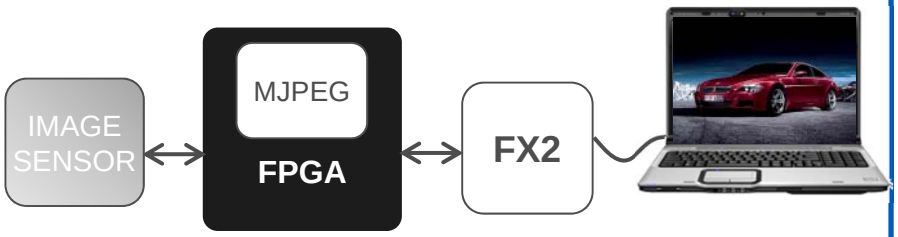
Video Compression

USB 2.0, NO COMPRESSION



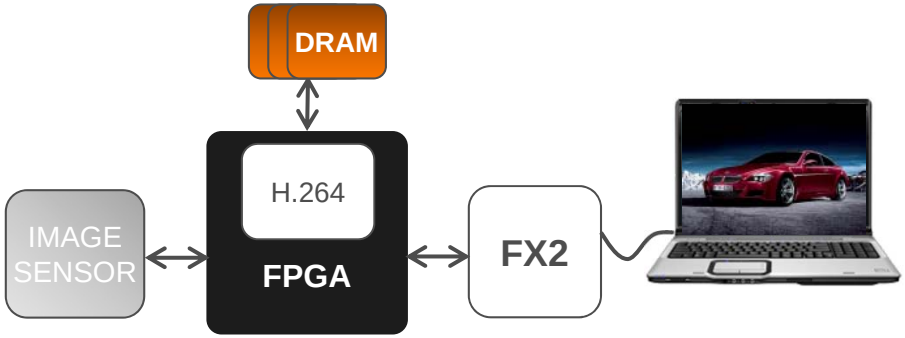
- LIMITED VIDEO QUALITY, LOW FRAME RATE

USB 2.0, MJPEG COMPRESSION



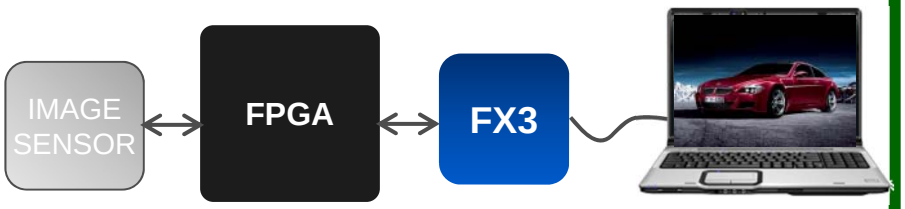
- LOW COST IMPLEMENTATION
- GOOD ENOUGH IMAGE QUALITY BUT LOSSY

USB 2.0, H.264 COMPRESSION



- HIGHER BOM COST AND MORE BOARD SPACE
- BETTER IMAGE QUALITY FOR MOST APPLICATIONS

USB 3.0, NO COMPRESSION



- LOWER BOM COST AND SMALLER BOARD
- HIGHEST IMAGE QUALITY

FX3 Overview: Target Applications

CONSUMER CAMERAS

GAMING

INDUSTRIAL CAMERAS

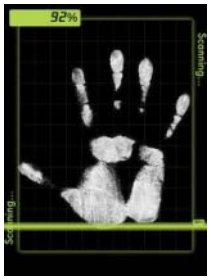
MEDICAL IMAGING

SURVEILLANCE

VIDEO CONFERENCING

SCANNERS

BIOMETRICS



HD VIDEO CAPTURE

DATA ACQUISITION

AUDIO MIXER

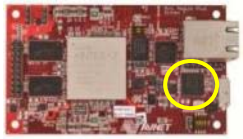
INSERT YOUR DESIGN HERE

PARTNERSHIPS TO HELP YOU SUCCEED

1. BUILDING STRONG USB 3.0 REPUTATION

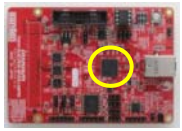
Microsoft USB Test Tool (MUTT) using FX3

2. WORKING WITH DISTRIBUTION AND PARTNERS



Avnet Kintex 7 Board Using FX3*

Arrow/Nuvation Board Using FX3*



Tokyo Electron Devices Board Using FX3*

3. BROADENING EVALUATION CHOICES

Braintechnology FX3 Mezzanine Board*

Enclustra FX3 Xilinx Adapter Board*

4. CREATING REFERENCE DESIGNS

Aptina "Demo 3" using FX3*

Net Vision SVI-06 Image Sensor Adapter Board*

*** ORDER BOARDS DIRECTLY FROM DISTRIBUTOR, SUPPLIER**

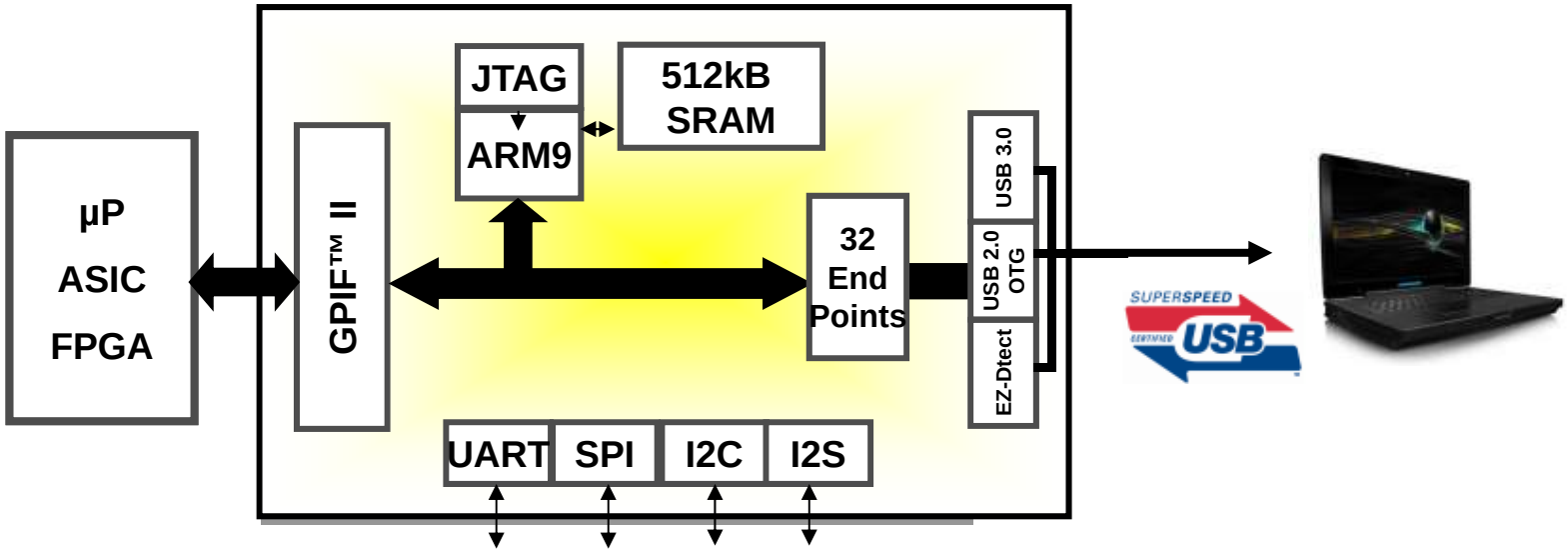
What is FX3?

Add USB3.0 capability to any system

- General Programmable Interface (GPIF™ II) to any processor/ASIC

Enable fast data processing and customization

- Fully accessible 32-bit ARM9 core with 512kB of embedded SRAM



FX2LP vs. FX3

Feature	EZ-USB FX2LP	EZ-USB FX3
USB Controller	USB 2.0	USB 3.0 USB 2.0 OTG
Data Processing Capability	8051	ARM926
GPIF – General Programmable Interface	16-bit	32-bit
Clock Rate	48 MHz	100 MHz
No. of States Supported	8	256
Programmability	GPIF Designer	GPIF Designer II

10x Performance, With The Same Familiar Architecture

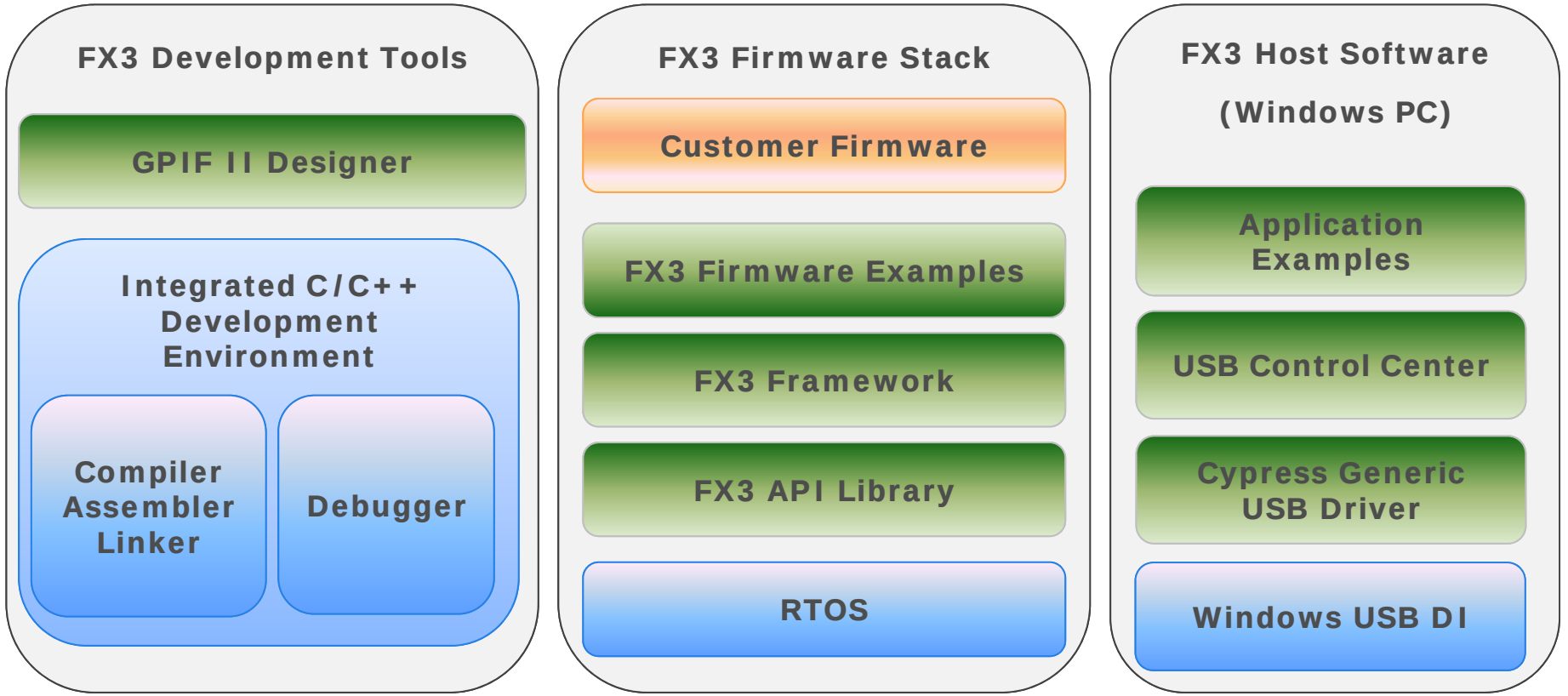
FX3 Booting

- FX3 can be the main processor or co-processor in a system
- PMODE[2:0] Configures the booting options
- FX3 booting options

PMODE[2:0]	Boot Option
F00	Sync ADMUX (16-bit)
F01	Async ADMUX (16-bit)
F11	USB Boot
F0F	Async SRAM (16-bit)
F1F	I^2C* (on failure, USB Boot)
1FF	I^2C* Only
0F1	SPI (on failure, USB Boot)

* If EEPROM is used for I^2C booting, 1 kOhm pull up resistors should be placed on the SCL and SDA lines for 1 MHz EEPROM communication.

FX3 SDK Components



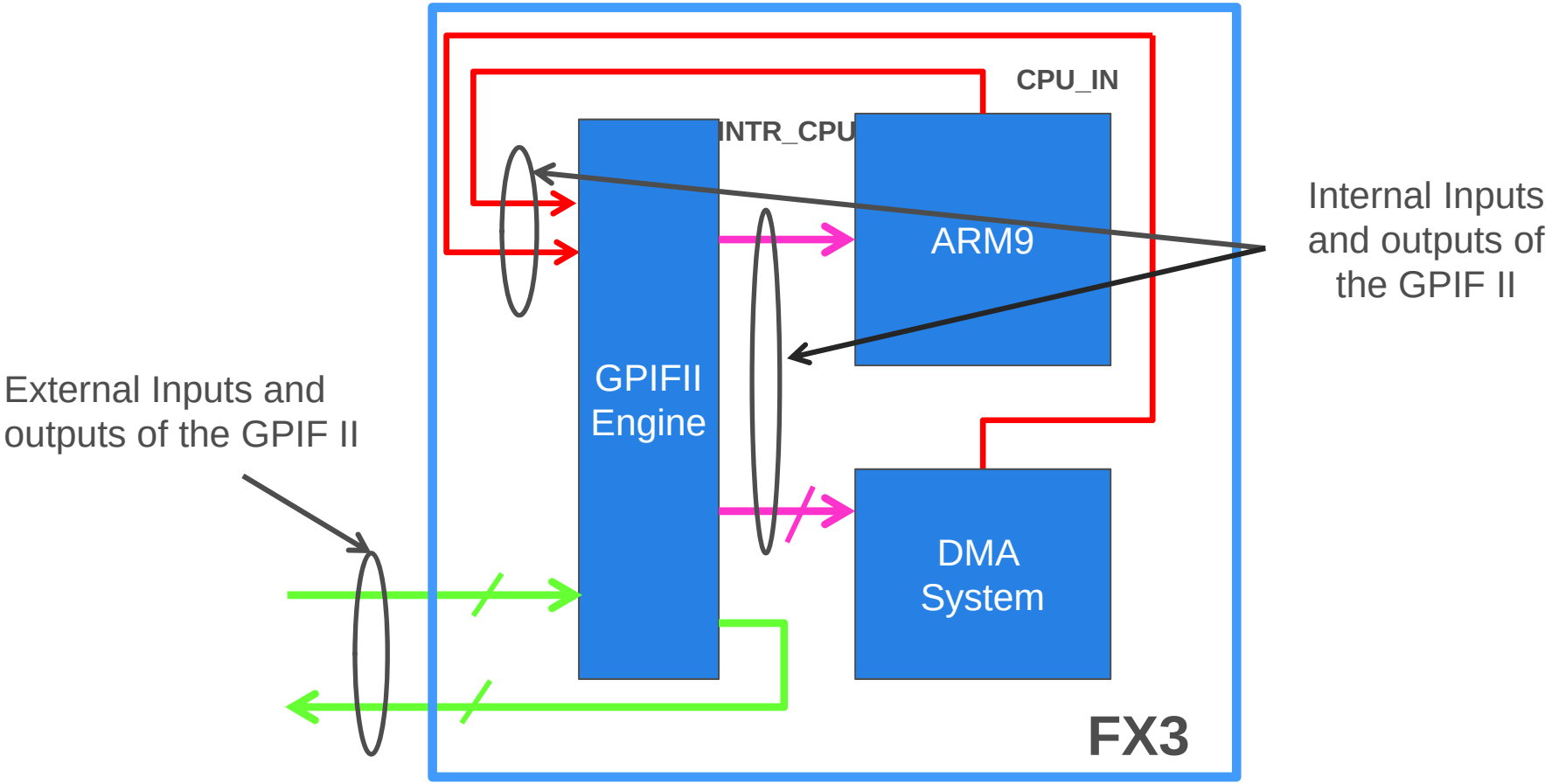
- Customer Software
- Cypress provided Software
- Third Party Software

Available Today!

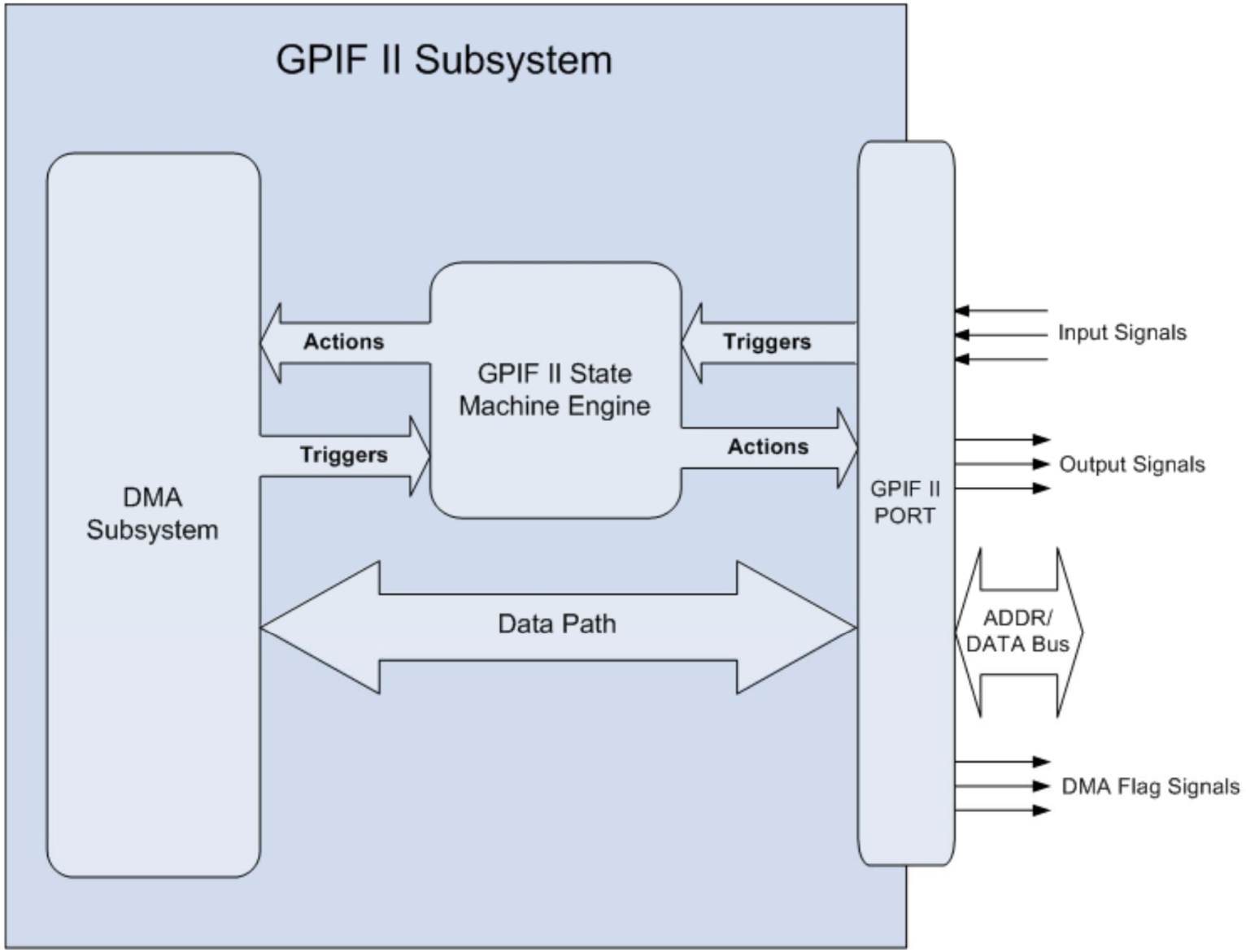
http://japan.cypress.com/?rID=57990

GPIF II Hardware: Overview

Following diagram depicts the concept of internal and external inputs/outputs of the GPIF II Engine



GPIF II Hardware: Overview



GPIF II Hardware: Signal Types

- **Alphas**
  - Outputs that depend on the transition
  - 8 alphas (4 user defined)
  - Example: sample_din
- **Betas**
  - Outputs that depend on the state
  - 31 Betas (4 user defined)
  - Example: data_count_incr
- **Lambdas**
  - Inputs that go into transition equations
  - 32 Lambdas (16 user defined)
  - Can use 4 per transition
  - Example: data_comp

GPIF II Hardware: Control signals

- **OE: Output Enable**
  - Provides control over the output drivers that determine if the bus is driven or tristate
- **WE: Write Enable**
  - Disables output drivers
- **DLE: Data Latch Enable**
  - Allows control over latching data from the data bus
  - Normally combined with WE
- **ALE: Address Latch Enable**
  - Allows control to latch address values
- **DRQ: DMA Request**
  - Provides a DMA request output that is controlled by state machine and responds to DACK signal
- **DACK: DMA Acknowledge**
  - Not a separate function but is used to control behaviour of DRQ

GPIF II Hardware: CPU Interrupts

- **Triggered by INTR_CPU action and other GPIF events**
- **Handled in firmware with callback function**
  - `Void CyFxGpifCB (CyU3PGpifEventType event,uint8_t currentState)`
- **INTR_CPU generates a `CYU3P_GPIF_EVT_SM_INTERRUPT` event**
- **Passes a currentState parameter which indicates the state of the State Machine**
  - State numbers are assigned to state names from GPIF II designer in H file

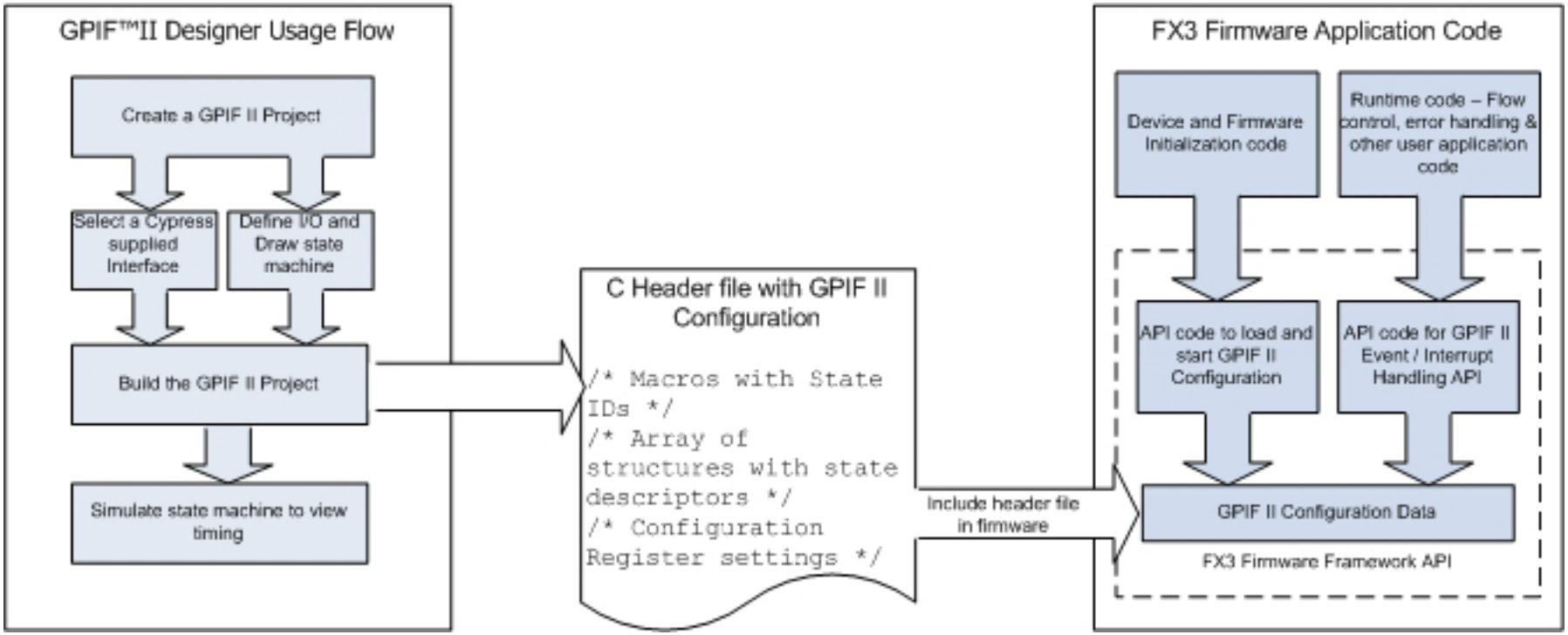
GPIF II Hardware: Counters

- **Implemented in GPIF II hardware**
- **Can be reset or updated by state machine**
- **Can generate limit or match signals as a trigger**

- **Address Counter**
  - Can check if the counter matches the current address
- **Control Counter**
  - Can check if the counter matches the current control inputs
- **Data Counter**
  - Can check if counter matches current data

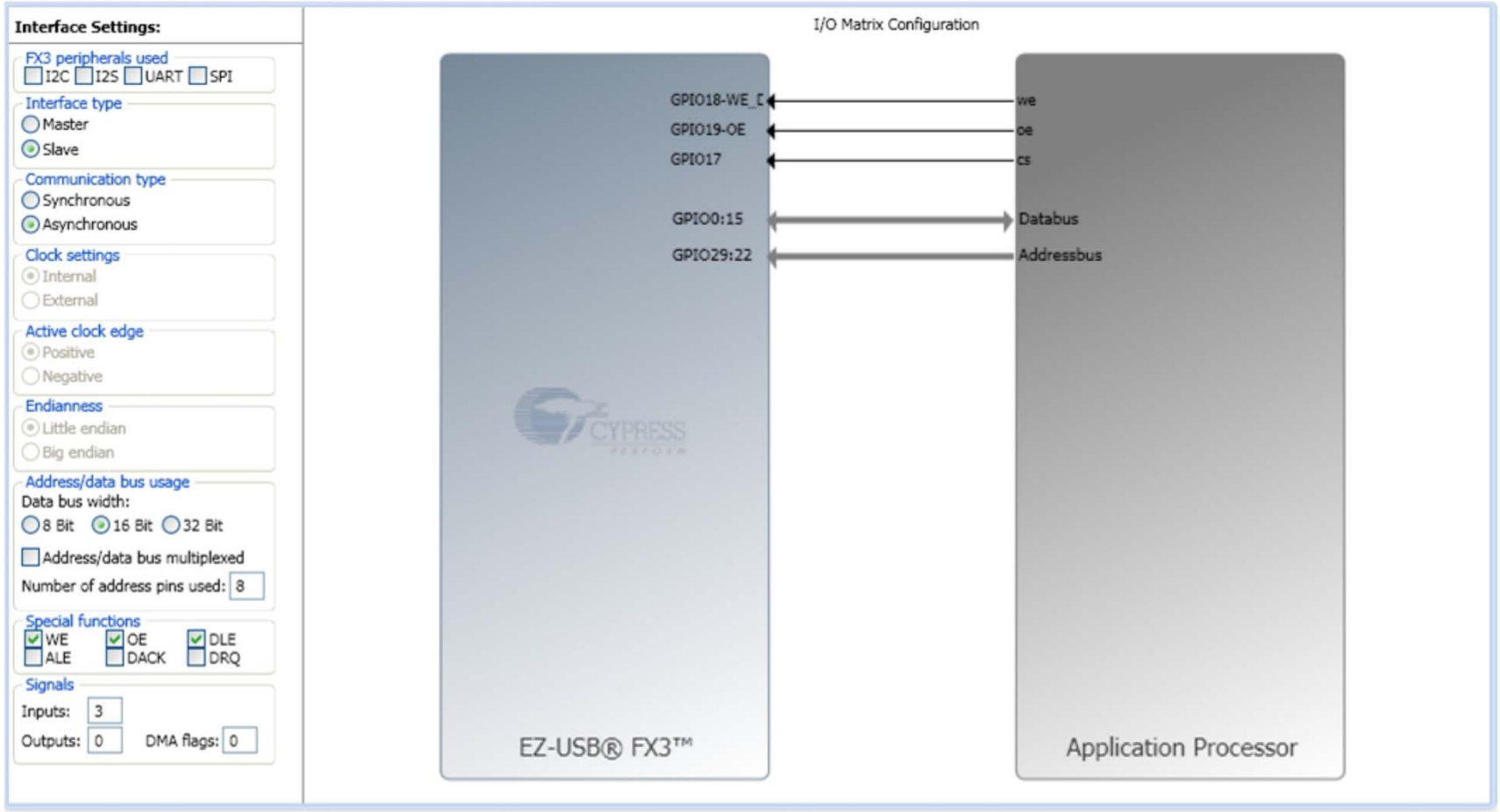
GPIF II Designer: Design Work Flow

1. **Design State Machine**
2. **Setup GPIF II interface in GPIF II Designer**
3. **Create State Machine in GPIF II Designer**
4. **Build design into C header file**
5. **Simulate State Machine**
6. **Integrate design into C firmware application code**



GPIF II Designer: Supplied interfaces

- **7 Cypress supplied interfaces**
- **Even with supplied interface it is still flexible**

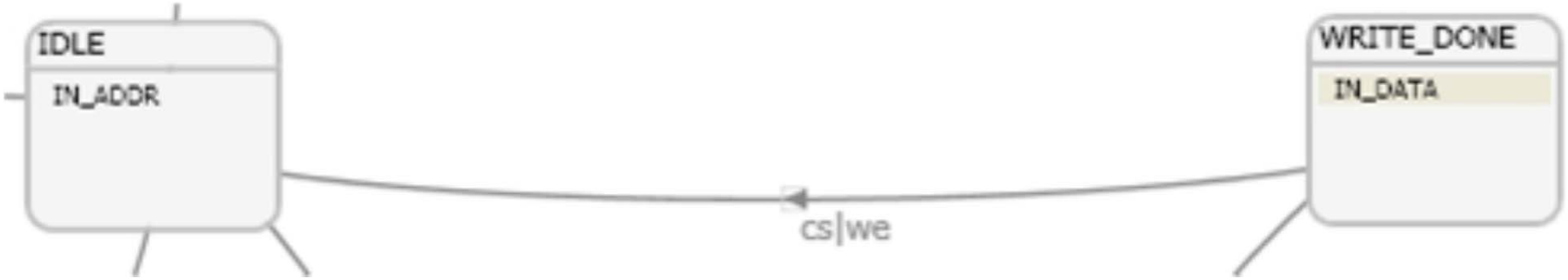


GPIF II Designer: Interface Settings

- **Peripherals**
  - Enabling I2C, SPI, UART, and I2S will have effect on GPIF II port
  - The peripheral selection is only used by the tool to calculate the available pin count. User is responsible for setting up the FX3 I/O Matrix in the firmware application code.
- **Master and Slave**
  - Initiate transfer or respond to a transfer
- **Interface clocking**
  - Synchronous or Asynchronous
  - Internal or External
- **Control Signals and GPIOs**
  - GPIF II state machine inputs/output
  - Unused control signals can be override by firmware
- **DMA flags**
  - Special hardware flags controlled by DMA engine not GPIF II

GPIF II Designer: Custom Interfaces

- **State machine canvas allows you to drag and drop states and transitions**
- **Transition based upon internal GPIF II signals as well as user defined signals**
- **Actions are macros that simplify GPIF II outputs**



GPIF II Designer: Actions

Actions control internal and external outputs

Relevant actions

- **LOAD_DATA_COUNT**
  - Sets up a hardware counter
  - Can be configured in a variety of ways
  - We will use it to keep track of data in our buffer
- **COUNT_DATA**
  - Increment data counter
- **IN_DATA**
  - Sample data from data bus and send it to a socket or register
- **INTR_CPU**
  - Generates CPU interrupt to alert FX3
  - FX3 can query which state the GPIF is in when it receives the interrupt

Details about more actions can be found in the GPIF II guide

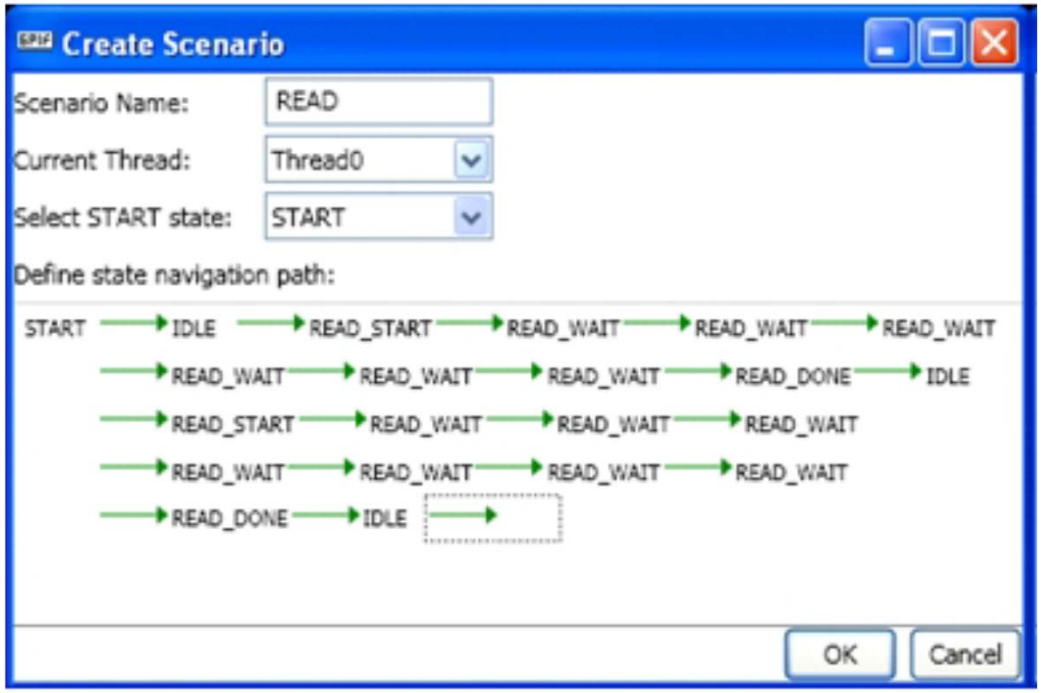
GPIF II Designer: API Calls

- **CyU3PGpifLoad**
  - Load GPIF II waveform descriptor into FX3 memory
- **CyU3PGpifSMStart**
  - Start GPIF II from initial state
- **CyU3PGpifSMSwitch**
  - Transitions state machine into a new state without stopping the machine
- **CyU3PGpifSMControl**
  - Pause and Resume GPIF II

GPIF II Designer: Simulation

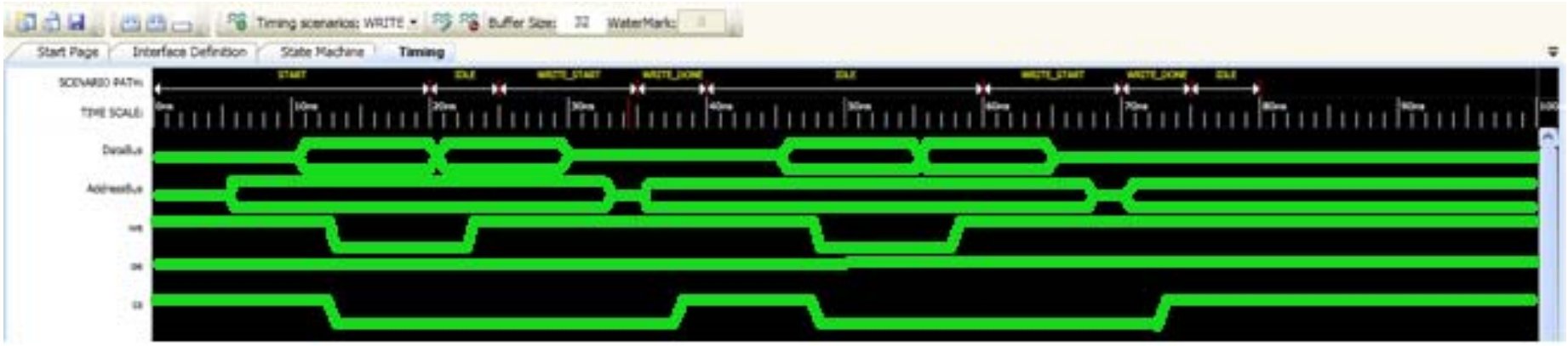
State Machine Path:

The path that is followed through the state machine due to a series of inputs



GPIF II Designer: Simulation

- **Define State Machine Path not inputs**
- **GPIF II Designer creates input signals based on path**
- **Displays output signals**
- **Determines Current State**



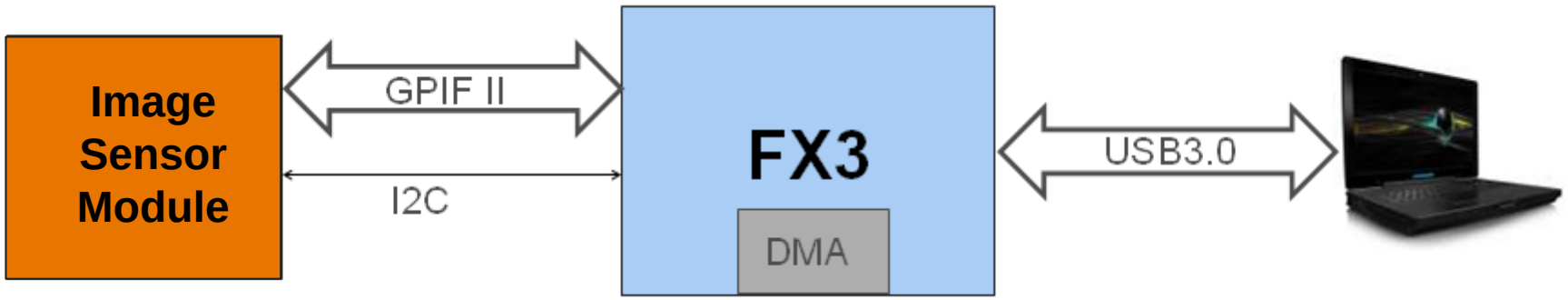
Tour of GPIF II Designer

Tour completed outside of powerpoint

http://www.cypress.com/?rID=62036

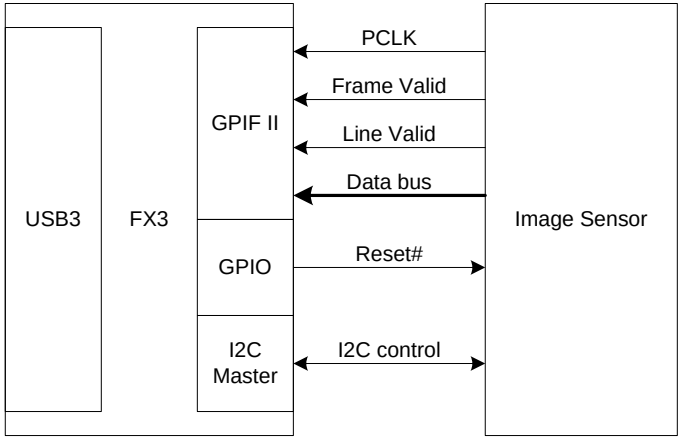
Lab 1 : Objectives

- **Setup IO configuration in GPIF II Designer**
- **Layout complex state machine in GPIF II Designer**
- **Run timing simulation in GPIF II Designer**
- **Build GPIF II Designer project**
- **Integrate GPIF II header file into FX3 firmware**



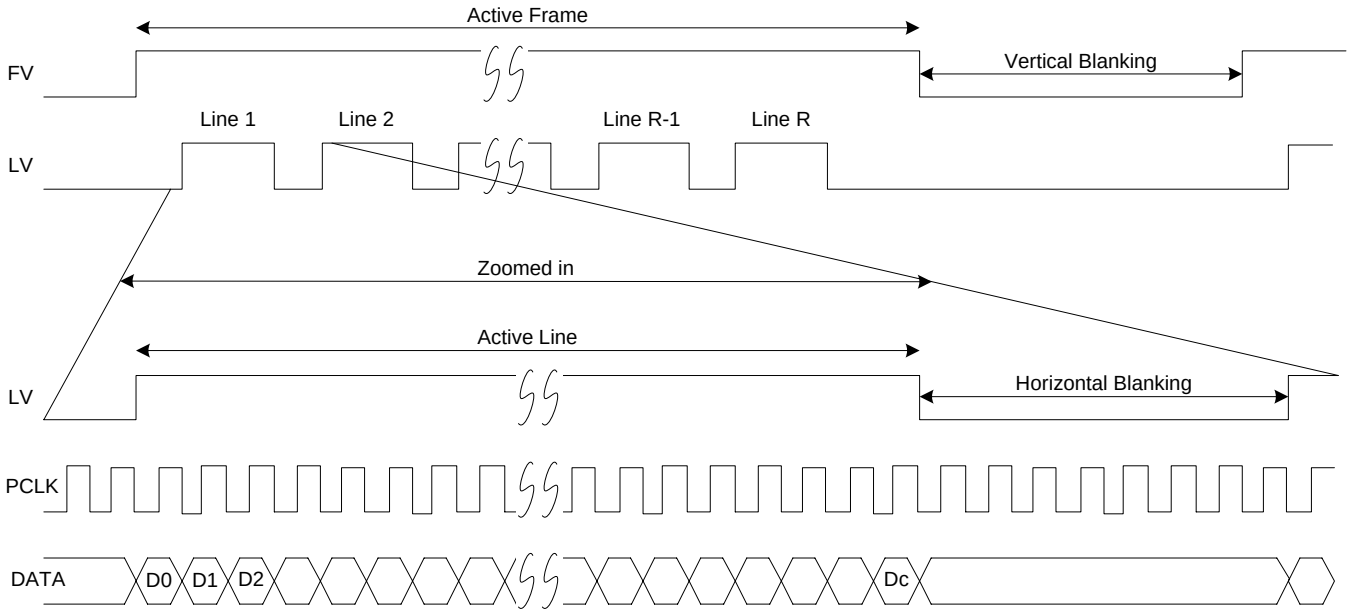
Lab 1: Physical Interface

- **3 inputs**
  - Handled by GPIF II so no firmware intervention necessary on these signals
- **Data bus**
  - Goes directly into DMA through the GPIF II Hardware
- **1 output**
  - Handled in firmware not GPIF descriptor
- **I2C control lines**
  - Control image sensor configuration
  - Handled in firmware

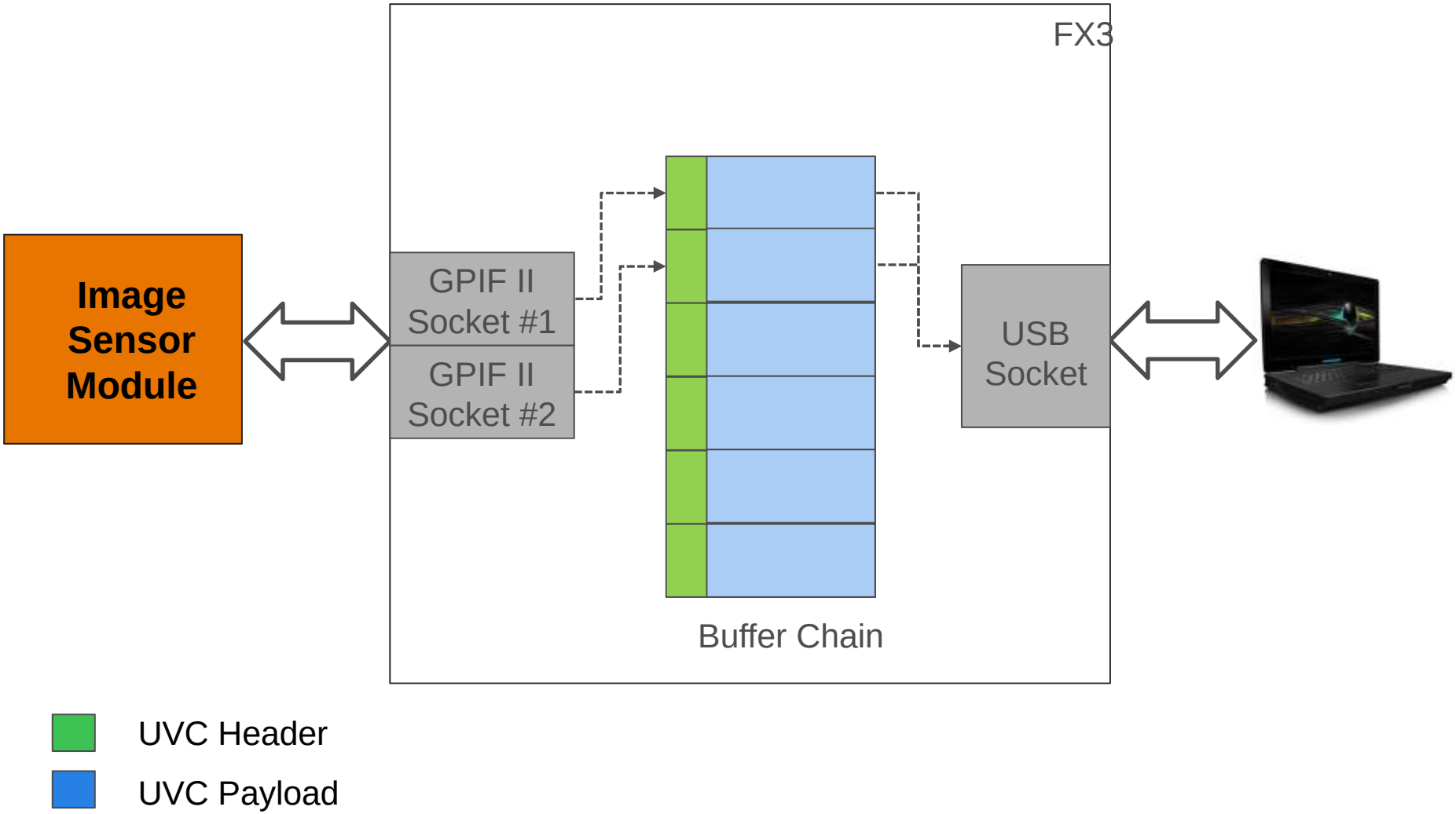


Lab 1: Timing Diagram

- **Frame valid (FV) high during active frame**
- **Line valid (LV) high during active line**
- **Pixel Clock (PCLK) always running**
- **PCLK can be used as clock source for GPIF II**
- **Data only valid when LV and FV are both high**
- **Data latched on positive clock edge of PCLK**

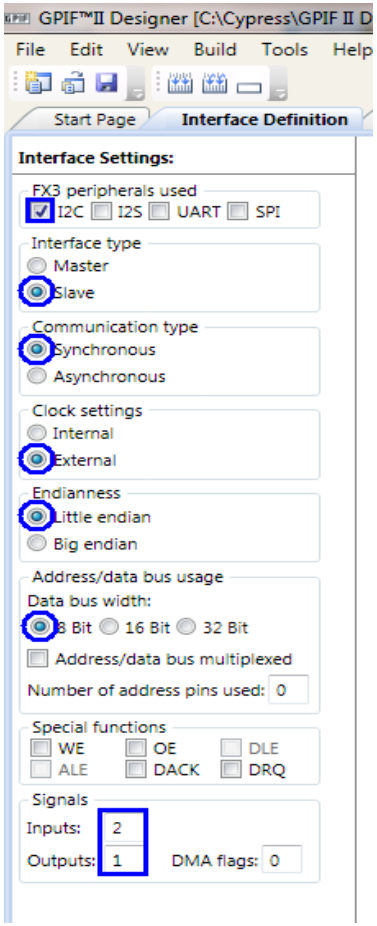


Lab 1: Design Considerations

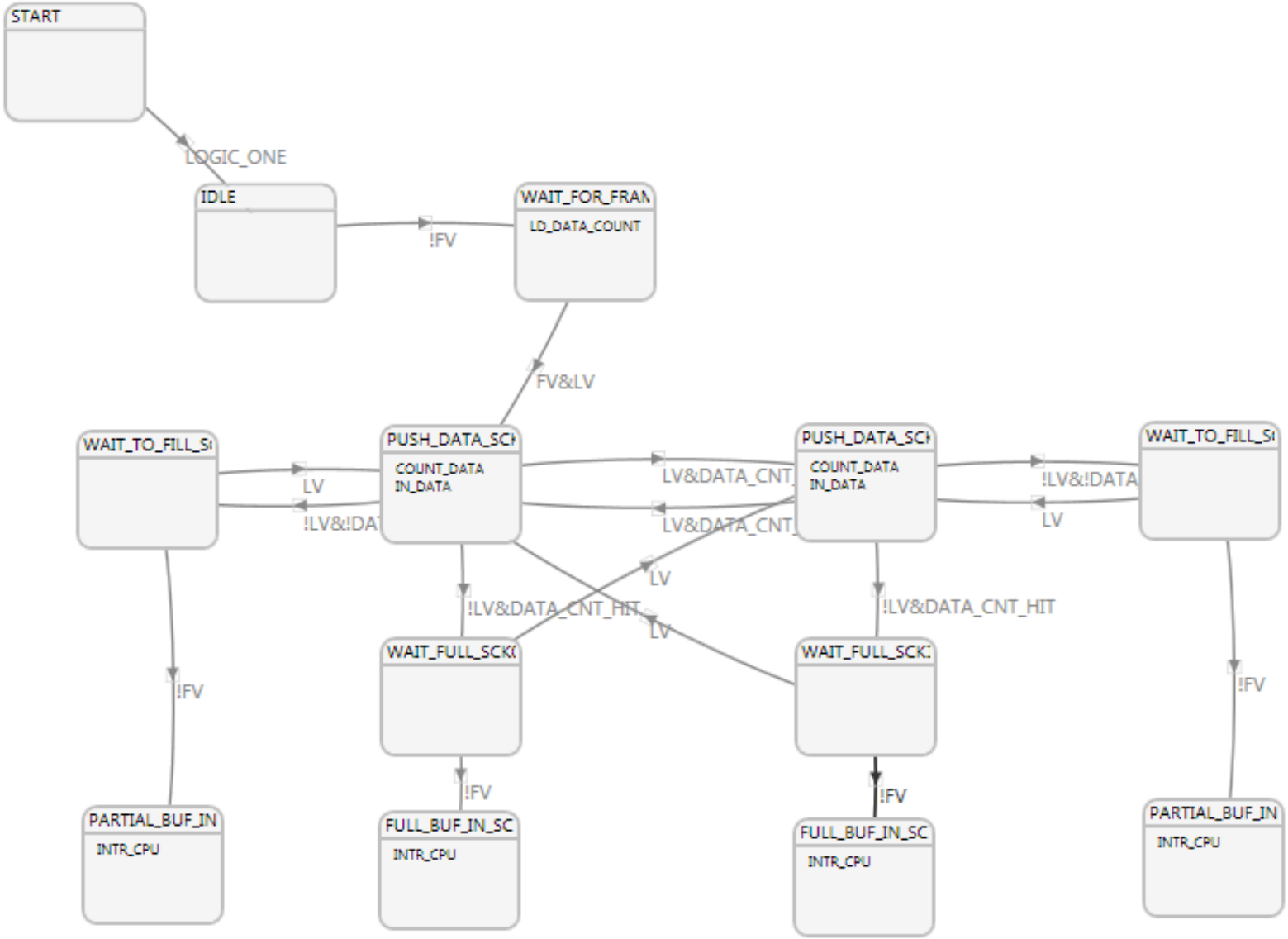


Lab 1: Interface Definition

- I2C enabled for communications
- FX3 is in a slave role
- Synchronous because we will be using a clock
- External Clock source since clock is coming from image sensor
- Data format is little endian out of image sensor
- 8 bit data bus width from image sensor with no address lines
- 2 input: FV, LV
- 1 output: nReset



Lab 1: State Machine



Collaterals

FX3 Data Sheet

Application Notes

- AN68914 - EZ-USB FX3 I2C Boot Option
- AN73150 - Booting EZ-USB FX3 over High-Speed USB
- AN73304 - Booting EZ-USB FX3 over Synchronous ADMux Interfaces
- AN70193 - EZ-USB FX3 SPI Boot Option
- AN70707 - EZ-USB FX3 Hardware Design Guidelines and Schematic Checklist
- AN65974 - Designing with the EZ-USB FX3 Slave FIFO Interface
- AN68829 - Slave FIFO Interface for EZ-USB FX3: 5-Bit Address Mode
- AN75705 - Getting Started with FX3
- AN75432 - USB 3.0 EZ-USB FX3 Orientation
- AN76348 - Migrating from EZ-USB FX2LP Based Design to EZ-USB FX3 Based Design

Collaterals

Videos

- Cypress EZ-USB FX3 Technology Overview
- EZ-USB FX3 Maximum Throughput Demo
- GPIF II Designer Introduction

Software Development Kit

FX3 DVK


CYPRESS
PERFORM