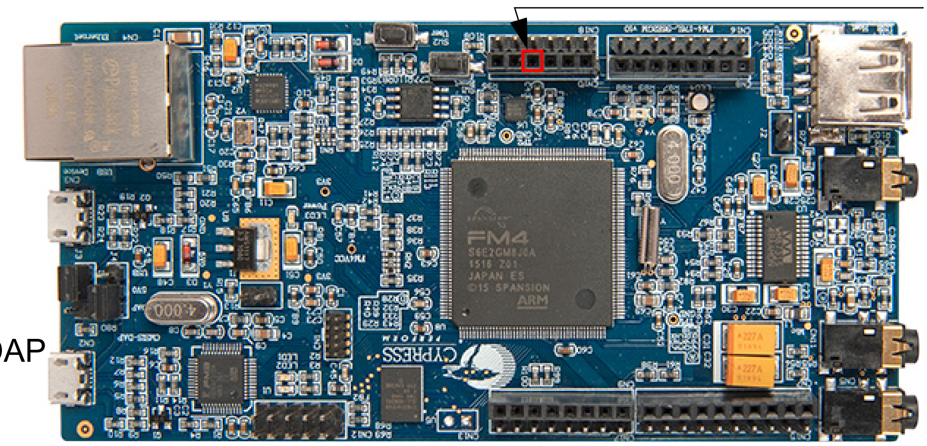
Real-Time DSP

ECE 5655/4655 Lecture Notes

P10/A3 Dig. Out



Line In

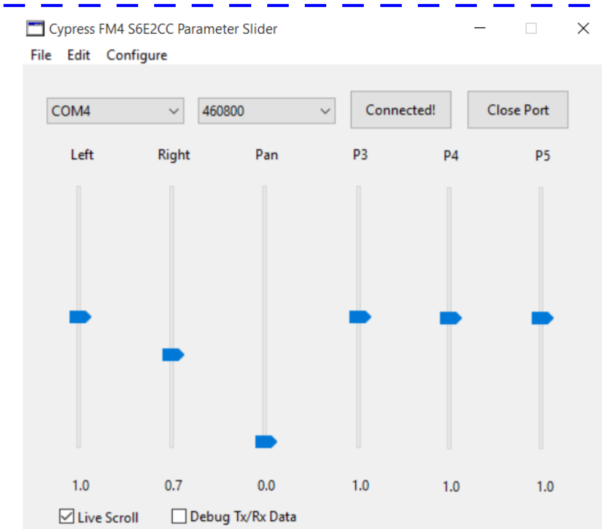
Mic In

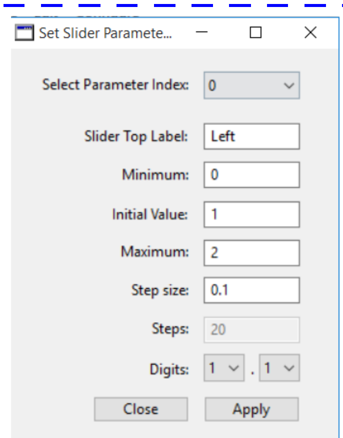
Headphone/ Line Out

CMSIS DAP & Virtual Serial

Cypress FM4 S6E2CC Pioneer Kit

Real-Time Parameter Control for the FM4





Cortex M4 Edition with Emphasis on the Cypress FM4

Mark A. Wickert

Chapter 1

Overview of Real-Time Digital Signal Processing

Introduction

In this first chapter we provide motivation for the topics to be addressed in this course. Before going any further let us first give a short description of the course and the assumed background for the course

A Brief Description of the Course

- A course in real-time DSP brings together the following:
  - Continuous- and discrete-time systems theory (in particular knowledge from a first DSP course)
  - Software engineering concepts
  - Microprocessor programming and hardware interfacing
- The interest in doing this stems from the increase in real-time DSP applications headed for the consumer market, and the ever improving device VLSI designs for implementing powerful DSP microprocessors

- Many years ago, Texas Instruments created the following long list of DSP real-time application areas

Automotive	**Consumer**	**Control**
Adaptive ride control	Digital radios/TVs	Disk drive control
Antiskid brakes	Educational toys	Engine control
Cellular telephones	Music synthesizers	Laser printer control
Digital radios	Pagers	Motor control
Engine control	Power tools	Robotics control
Navigation and global positioning	Radar detectors	Servo control
Vibration analysis	Solid-state answering machines	
Voice commands		
Anticollision radar		
General-Purpose	**Graphics/Imaging**	**Industrial**
Adaptive filtering	3-D rotation	Numeric control
Convolution	Animation/digital maps	Power-line monitoring
Correlation	Homomorphic processing	Robotics
Digital filtering	Image compression/transmission	Security access
Fast Fourier transforms	Image enhancement	
Hilbert transforms	Pattern recognition	
Waveform generation	Robot vision	
Windowing	Workstations	
Instrumentation	**Medical**	**Military**
Digital filtering	Diagnostic equipment	Image processing
Function generation	Fetal monitoring	Missile guidance
Pattern matching	Hearing aids	Navigation
Phase-locked loops	Patient monitoring	Radar processing
Seismic processing	Prosthetics	Radio frequency modems
Spectrum analysis	Ultrasound equipment	Secure communications
Transient analysis		Sonar processing
Telecommunications		**Voice/Speech**
1200- to 33 600-bps modems	Faxing	Speaker verification
Adaptive equalizers	Line repeaters	Speech enhancement
ADPCM transcoders	Personal communications systems (PCS)	Speech recognition
Cellular telephones	Personal digital assistants (PDA)	Speech synthesis
Channel multiplexing	Speaker phones	Speech vocoding
Data encryption	Spread spectrum communications	Text-to-speech
Digital PBXs	Video conferencing	Voice mail
Digital speech interpolation (DSI)	X.25 packet switching	
DTMF encoding/decoding		
Echo cancellation		

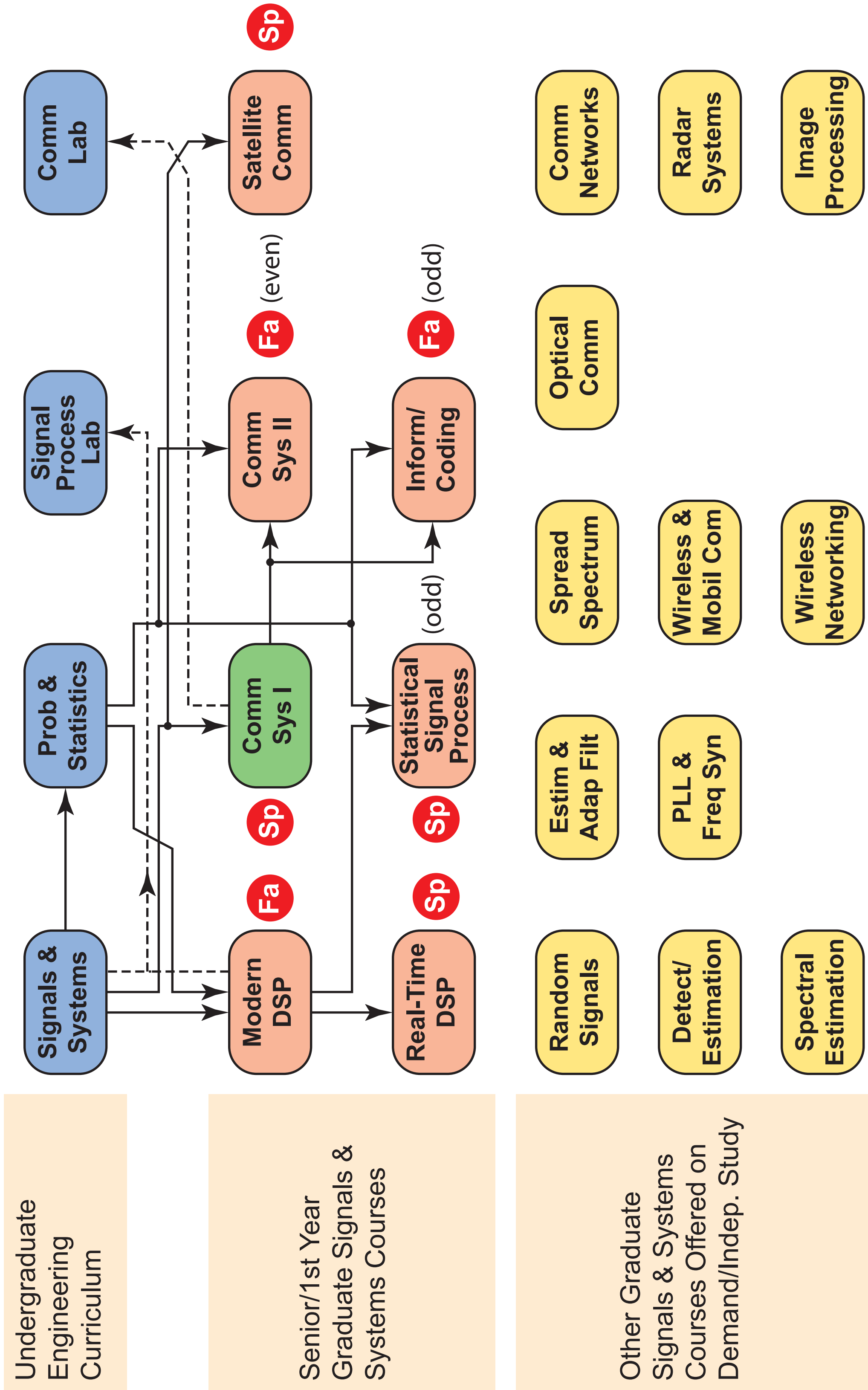
- Today (2017) the list is longer, with many of the DSP applications no longer a going concern, e.g., 1200 to 36600-bps modems
  – Do you remember using telephone channel modems, or too young?

- This course is about the use of general purpose digital signal processing microprocessors for solving signal processing problems in real-time
- The course focus will be on using the ARM Cortex M series, specifically the M4 processor for general purpose DSP
- This is a huge departure from the past 10+ years where this course has been taught using the Texas Instruments (TI) C6x family fixed/floating processors
- The course will start out considering general signal processing applications of real-time DSP and the associated programming issues
  - Later in the course, a focus applications area will be wireless communication system design using DSP algorithms
- The book chosen for the course
- The course meeting time will be used for lecturing and laboratory time (about 60/40) using hardware/software development tools
  - Tools: MDK–ARM development environment for Cortex and ARM devices
  - Specifically MDK-ARM v5.x (also denoted as μVision) is a product of the ARMKEIL microcontroller tools group in Germany

Background Requirements

- The required background for all students taking the course is an introductory graduate or junior/senior level undergraduate course in DSP and experience programming in ANSI C
- Knowledge of ANSI C is required in order to develop real-time algorithm in a high level language (HLL)
- At this point it is unlikely that we do any assembly language programming (I'm looking at my options in this area)
- Eventually, and in practice, you use a combination of C and assembly, e.g., mixed language programming
- That said, for the Coretex M we have at our disposal the CMSIS-DSP
- Programming the PC host in C/C++, or perhaps some other language, may be useful for developing a user interactive application for the final project
- Familiarity with test equipment e.g., signal generators, digital scopes, and a spectrum/network analyzer would be helpful

Course Perspective



A Brief History of DSP in the Context of Real-Time Processing

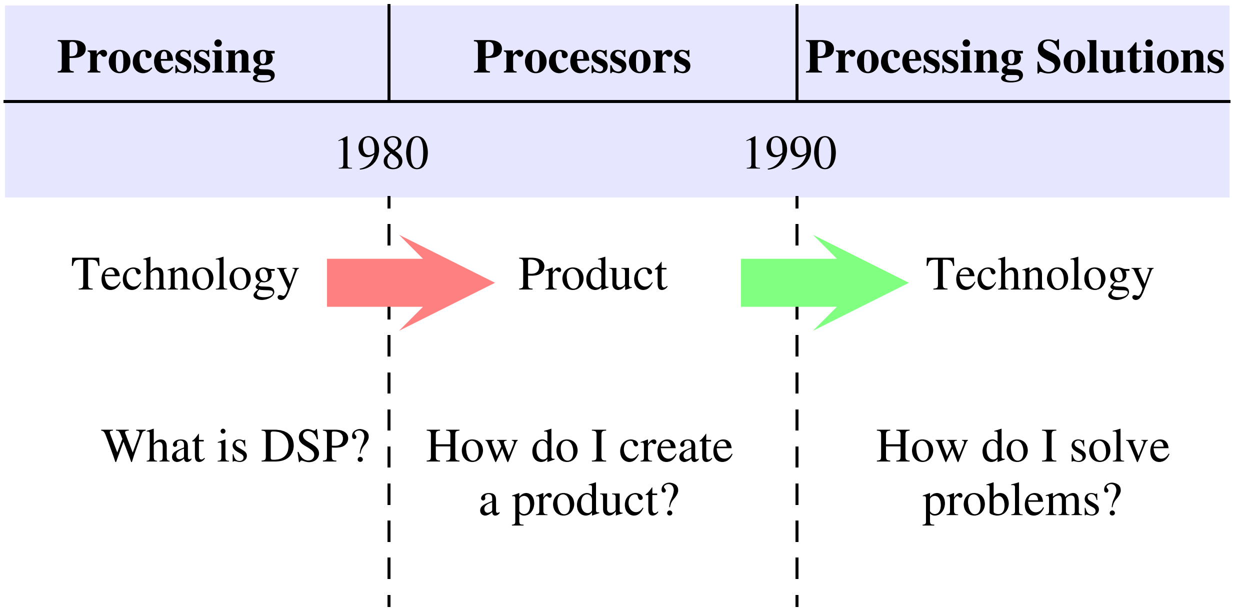
- From a systems engineering point of view DSP stands for digital signal processing, but in the world of real-time hardware, systems people often refer to digital signal processing solutions (DSPS)
- The decades of DSP/DSPS have brought the following[1]

Table 1.1: Decades of DSP

Decade	Characteristic	$/MIPS
‘60s	University Curiosity	$100–$1,000
‘70s	Military Advantage	$10–$100
‘80s	Commercial Success	$1–$10
‘90s	Consumer Enabler	$0.10–$1
Beyond	Expected Part of Daily Life?	$0.01–$0.10

1. Gene A. Frantz, “TI’s DSPS Future,” *Texas Instruments DSPS Fest ‘97*, Houston, TX, July 1997.

- Another way of looking at this is (Frantz)



- We are well beyond the time line shown above, what has been happening in the last 12 years?
  - DSP is definitely mainstream in most everything electronic
  - general purpose DSP processors are only part of the picture; FPGA drives the government/military applications side, and ARM processors drive the consumer product side
  - In the last few years we have also seen more multi-core devices

Great Moments in DSPS History[1]

- 1976 – DSP is used to simulate a voice in the educational product "Speak and Spell"
- 1982 – TI announces details on the TMS32010 which executes at 5 MIPS
- 1985 – DSP is used in a modem for the first time
- 1986 – Lotus automotive uses DSP in active suspension and noise abatement in racing cars
- 1988 – First DSP hearing aid introduced
- 1991 – First TI sponsored educators conference
- 1993 – Cadillac automotive introduces a DSP-based ride control system
- 1995 – TI implements the *On-line DSP Lab™* for Web based testing of DSP applications
- 1997 – 15 years of DSPs for TI and
  - The introduction of the TMS320C6x family of DSPs 1,600 MIPS performance (C62x)
  - The 1-v barrier in power consumption reached
  - In late 1997 the first C6x floating point part is announced (C67x) with 1 GFLOPS performance
- 2012?

1. "Great Moments in DSPS History," *Integration: An Update on Texas Instruments Semiconductors*, vol. 14, No. 4, June 1997.

DSP Integration (Frantz)

Table 1.2: DSP Integration over three decades.

	Typical 1982 DSP	Typical 1992 DSP (97)	Typical 2002 DSP
Die size	50 mm	50 mm	50 mm
Technology size	3 μ	0.8 (.35) μ	0.18 μ
MIPS	5	40 (100)	2000
MHz	20	80 (200)	500
RAM (words)	144	1 K	16 K
ROM (words)	1.5 K	4 K	64 K
Price	$150.00	$15.00	$1.50
Pwr. Dissp.	250 mW/ MIP	12.5 mW/ MIP (1)	0.1 mW/MIP
Transistors	50 K	500 K	5 M
Wafer Size	3” (75 mm)	6” (150 mm)	12” (300 mm)

- Where are we at in 2017?

Communications and Wireless

- In the past two key application areas, telephone line modems and cellular voice/data communications, have had a major impact on DSPS sales
- In 1994 when the V.34 modem standard (28.8 kbps) was approved fixed point (integer) DSPs became a popular design route
  - The V.34 standard allowed for data rates from 2400 to 28,800 bps
  - The modulation scheme at 28,800 is a high-order quadrature amplitude modulation (QAM) employing *shell mapping*, *trellis encoding*, *nonlinear precoding*, and *soft decision Viterbi decoding*[1]
  - All of the above and backward compatibility (lower rate schemes) can be performed by one 50 MIPS fixed point DSP
  - In late 1994 these modems cost ~$400, today we know that 28,800 modems (dial-up) are not used much at all
- In the cellular telephone arena fixed point DSPs are widely used in the portable handsets
- DSPs are also utilized in VoIP and cellular basestations
  - In a portable device market, such as cell phones, the effi-

1. Steven Tretter, *Communication System Design using DSP Algorithms with Laboratory Experiments for the TMS320C6713*, Plenum Press, New York, 2008.

ciency of a particular processor can be measured in mW/algorithm, which is obtained via

$$\frac{\text{mW}}{\text{algorithm}} = \frac{\text{mA}}{\text{MIPS}} \times \frac{\text{MIPS}}{\text{algorithm}} \times \text{voltage}$$

- MIPS/algorithm requirements for an IS95 handset[1]

Table 1.3: MIPS requirements for an IS95 handset.

Algorithm	Implementation	MIPS
Correlator	Hardware	5
Automatic frequency control (AFC)	Hardware	5
Automatic gain control (AGC)	Hardware	5
Transmit filter	Hardware	30
128-pt FFT	Software	1
Viterbi decoder (length 9, rate 1/2)	Software	6
Vocoder (8-Kbps Qualcomm code excited linear prediction (QCELP))	Software	20
Vocoder (enhanced variable rate coder (EVRC))	Software	30

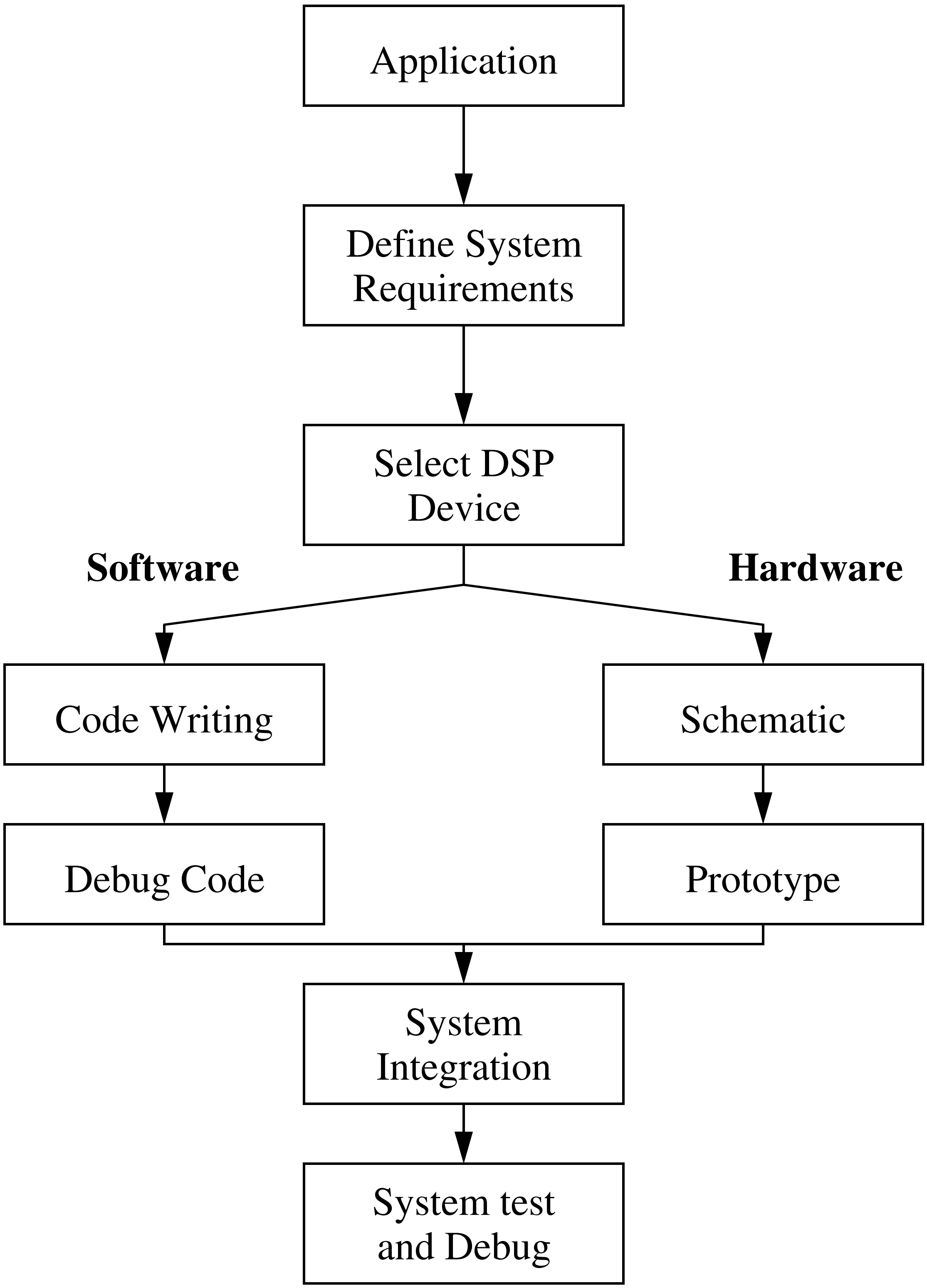
1. John Groe and Lawrence Larsen, *CDMA Mobile Radio Design*, Artech House, Boston, MA, 2000.

DSP Hardware Design Issues[1]

- Hardware alternatives, FPGA, ASIC
- Using a general purpose DSP
- Fixed-Point DSPS
  - Fixed-point arithmetic
  - Quantization effects and scaling in fixed point
  - Typically a lower power design
  - Typically a lower cost design
- Floating Point DSPS
  - A more rapid design due to the ease of floating point algorithm implementation
  - The Cortex M4 has one hardware multiply and accumulate function to allow rapid prototyping of algorithms
  - The CMSIS-DSP library includes fixed and floating point algorithms
- Accessing memory resources
- Integration of peripheral devices
- A very practical alternative today is to consider a hybrid FPGA/general purpose DSP architecture

1. Craig Marven and Gillian Ewers, *A Simple Approach to Digital Signal Processing*, John Wiley, 1996. ISBN 0-471-15243-9.

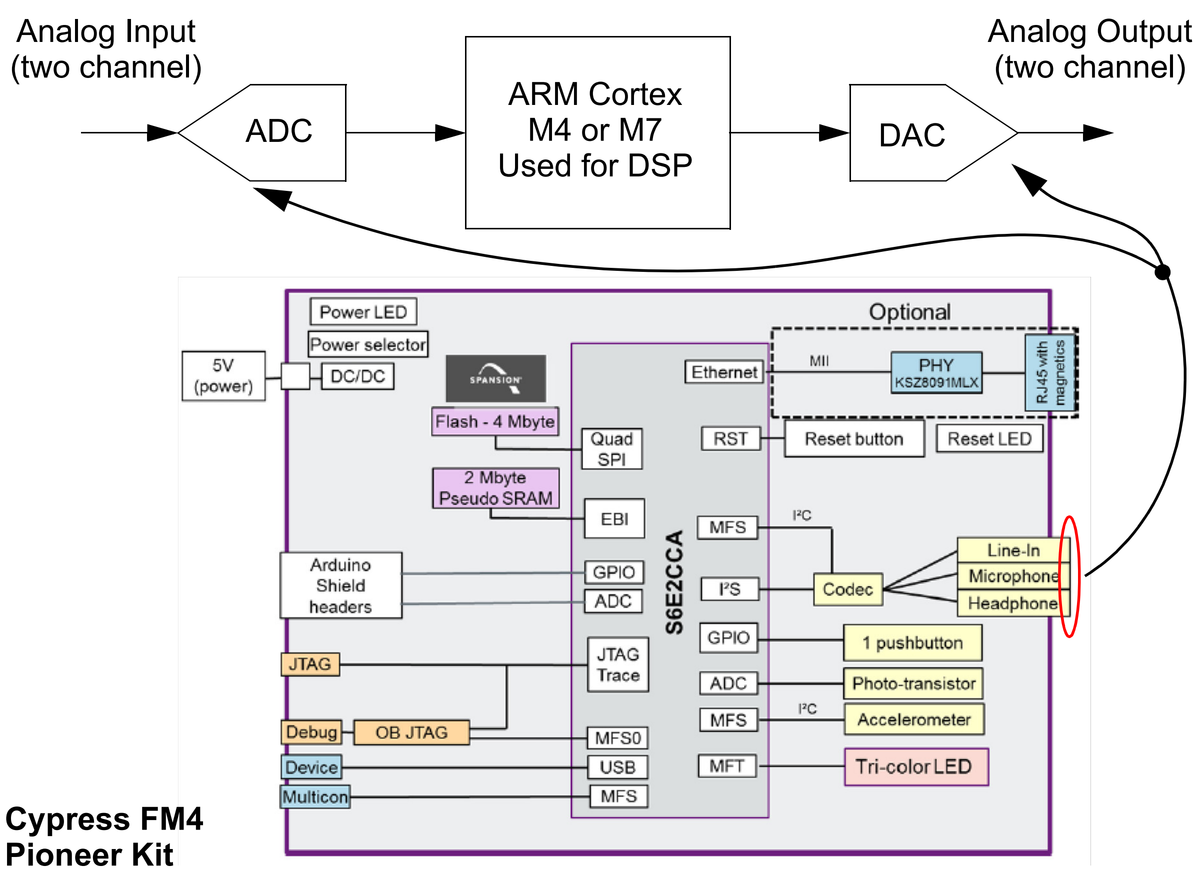
DSP System Design Flow (Marven & Ewers)



- Application Design
  - Software design
  - Generating assembler source and using C where possible
  - Testing the code
  - Hardware design
  - System integration

Systems Level M4 or M7 Hardware

- Many hardware/dev kit options available:
  - In 2015 the STM32F407 (M4) (~168 MHz)
  - In 2015 the NXP LPC4088 + Baseboard (M4) (~120 MHz)
  - TI Tiva TM4C123G (M4) + Audio Booster Pak (~80MHz)
  - From 2016 to the present Cypress FM4 (M4) (200 MHz)
  - The STM32F746 (M7) (~212 MHz) looked interesting in 2016, but did not take off as expected
  - Others? Yes!



Course Laboratory Foundations

- Beyond DSP math, you need to work with a variety of software and hardware, including measurement equipment

Software

- MDK–ARM (Keil) development platform
  - There are also open-source GCC based integrated development environments available for the M4
  - I am still thinking about this option, now also considering using PSoC 6 which has its own IDE (PSoC Creator and WICED Studio for BLE and WiFi together)
- TI integrated development environment *Code Composer Studio*;
  - The above IDE is based on *Eclipse*
  - The Reay text refers to an audio codec *BoosterPack* for the Tiva series of Cortex M4 processors; not readily available
- IPython and Jupyter notebook for all calculation needs
  - There is a learning curve to get going with PyLab and the IPython interface, but that fact that the tools are readily available is powerful motivation to *take the plunge*
- *GoldWave* or *Audacity* shareware for .wav audio file manipulation

Test Equipment

- For all of your measurement needs the Digilent *Analog Discovery* (or the new version 2) may be adequate (see in-class demo and syllabus for details)
  - With the Analog Discovery 2 you have a highly portable set of instruments with the software running on a PC/Mac/Linux
  - This would allow you to work from home
- Keysight DSOX6004A four-channel 1 GHz digital scopes (20 Gsps)
- Agilent function/arbitrary waveform generators; 15 MHz and 80 MHz models, and the new Keysight 33600A True*f*orm, two channel generators
- Agilent 10 Hz – 500 MHz spectrum/vector network analyzers
- For RF/microwave front-ends we have the Keysight FieldFox N9914A, 100 kHz – 6.5 GHz spectrum/vector network analyzers