TriMedia Foundation class

(typical schedule - individual classes may vary)

TriMedia Foundation class – daily class schedule

	Day 1	Day 2	Day 3	Day 4
	Software Architecture	System On Chip Architecture	CPU core architecture	Optimization
9.30 am	Preparation	Review questions on previous day's work		
10.00 am	System architecture	Basics of digital video	TriMedia CPU core architecture	Scheduling and Profiling
11.00 am	Software Architecture (TSSA 1.6)	TriMedia peripheral architecture	TriMedia compilation tools	Cache Optimizations
12.00 am	Morning review questions			
12.30 pm	lunch			
1.00 pm	Extension work			
1.30 pm	Software Architecture (TSSA 1.5)	pnx1300 System On Chip architecture	TriMedia cache architecture	Scheduling Optimizations
2.30 pm	Practical exercises, discussion and questions			
3.00 pm	Software Architecture (IADK / MPTK)	pnx1500 System On Chip architecture	TriMedia cache architecture	Custom and SIMD Optimizations
4.00 pm	Software Architecture (SDE2)	pnx1500 System On Chip architecture	Discussion	Certificate presentations
4.30 pm				
5.00 pm	Extension work			

Day 1: System architecture
Software Architecture

Day 2: System On Chip Architecture

Day 3: CPU core architecture

Day 4: Optimization

1. TriMedia System Architecture

Introducing the overall TriMedia system as an integrated System On Chip and Software Architecture: explaining its aims and the resulting design choices; outlining the main features of the overall system and of its software and hardware elements.

a. TriMedia System - an architecture for Media processing
b. Characteristics of media processing systems

c. The Universal Home API
d. TriMedia Software Architecture
e. Media Processing Toolkit

f. Architectural choices in media processing systems
g. Matching System on Chip to application needs
h. Mapping the application to the architecture
i. TriMedia core architecture
j. Compilation tools

k. Peripheral architecture

2. TriMedia Software Architecture

In-depth look at the TriMedia Software Architecture, including TSSA 1.6, 1.5 and 1.4 versions: outlining the history of its development and resulting legacy issues; explaining and clarifying the different programming models of different versions, and the intention and implementation of different layers within each version.

a. TSSA 1.6 software architecture

Introducing the TSSA 1.6 programming model including its choice of COM-like components and interfaces; explaining the concepts and implementation of component connection; and legacy issues for earlier (TSSA 1.5) components.

i. Components

TSSA 1.6 programming model of COM-like components and interfaces.

1. Contracts
2. Interfaces
3. Interface Suites
4. Interface names
5. Notifications

ii. tmCOM

Introducing the TSSA 1.6 COM implementation.

1. tmCom interface call
2. tmCom function tables
3. tmCom interfaces
4. tmCom creating components

iii. Connections

Explaining the concepts and implementation of connections in TSSA 1.6 including pins; queues and packet pools; the connection toolkit; and legacy issues for wrapping TSSA 1.5 components

1. Connection Manager
2. Component pins
3. Packet queues
4. Packet pools
5. Packet pool streaming to a queue
6. Packet streaming from a queue
7. Connection toolkit
8. Streaming interfaces
9. Connection sequence
10. Data Packets
11. Clocks
12. Synchronization
13. Creating components
14. TSSA 1.5 wrapper
15. TSSA 1.5 wrapper creating a component
16. TSSA 1.5 wrapper setting up a component

b. TSSA 1.5 software architecture

Legacy TSSA 1.5 software architecture, including Operating and Application Layers; InOutDescriptors as connecting components; and examples of usage.

i. TSSA 1.5 Software Architecture
    1. TSSA history
    2. Software Architecture Layers
    3. Software Architecture modes

ii. Operating Layer
    1. Software Components - OL
    2. TSA OL example
    3. TSA OL example - component
        a. TSA OL component configuration functions
        b. TSA OL component connection
        c. TSA OL component connection checking
        d. TSA component callback functions
        e. TSSA 1.5 component connection
    4. Data packets
        a. Data formats
        b. Video data format
        c. Audio data format
    5. IADK example - exolViVo
        a. IADK example - exolViCopyIOVo

c. Integrated Appliance Developer's Kit

Introduction to the IADK and MPTK component collections, through examples of mini-applications showing different aspects of usage.

i. IADK example - video encode and decode system
    1. a test harness for video codecs
ii. IADK example - exolVCodec
iii. DV decoder (digital video camera)
    1. components with multiple inputs and outputs
iv. MPEG-2 decoding
    1. dynamic disconnection and reconnection
v. MPEG-2 transport stream
    1. in-place components
vi. MPEG-4 decoding
    1. Audio / video synchronization

vii. pnx1500 peripheral components

Components for pnx1500 peripherals, including VIP and QVCP.

1. exolVrendVip example
2. exolVrendVip configuration
3. VcapVip component
4. pnx1500 video digitizer
5. VrendGfxVo component
6. pnx1500 video renderer

d. Operating System Abstraction

Concepts and implementation of Operating System Abstraction including legacy issues; and how pSOS+ is used.

i. OSAL background
ii. tmos
iii. tmosal
iv. pSOS+ operating system

e. Application Layer

Explanation of the purpose and use of the Application Layer including as a layer in which to write components that will be used at the OL; the purpose and implementation of the Default Layer; and how to use components at the Application Layer.

i. TSA AL components
ii. TSA AL example
    1. TSA AL example - component configuration
    2. TSA AL example - structures
    3. TSA AL example - open and close functions
    4. TSA AL example - data handling
    5. TSA AL example - processing
    6. TSA AL example - data formats
    7. TSA AL example -processing
iii. Default Layer

f. Software Development Environment

Aims and implementation of the Software Development Environment as a structure within which to develop and distribute re-usable components; including how to set up the SDE, and how to build applications and components.

i. SDE component folders
ii. SDE configuration
iii. SDE building
iv. Installing the TCS 4.3 software
v. Installing the TCS 4.4 software
vi. Building the MPTK software
vii. SDE configuration for UHAPI
viii. Making an application
ix. Making a component
x. Making an application component

3. Basics of digital video

Review of digital video including sampling schemes and storage formats. This is useful to cover common ground in preparation for sessions dealing with video peripherals.

a. The basis of 3-component color
b. Color spaces
    i. RGB
    ii. YUV
    iii. YCbCr
c. Raster scan
    i. Video frames
    ii. Video fields
        1. Interlacing
        2. Pull down
d. Video formats
    i. Video perceptive compression
    ii. YUV 4:2:2
    iii. YUV sampling schemes
e. Planar video formats
f. MPEG SIF
g. Video connectors

4. TriMedia Peripheral Architecture

Introduction to the TriMedia System on Chip peripheral architecture.

a. Peripheral architecture
b. Memory map
c. Peripheral MMIO registers
d. Software Architecture - devices
    i. Device library layer
    ii. Board Support Library
    iii. Peripheral registers configuration data structures
    iv. Peripheral registers configuration functions
e. Interrupts
    i. Interrupt configuration
    ii. Interrupt programming
    iii. Interrupts and decision trees
f. Data cache coherency and MMIO
    i. Data cache copyback example
g. Peripheral simulation

5. pnx1300 System on Chip architecture

In-depth look at the pnx1300 System on Chip architecture including pnx1300 peripheral architecture and the configuration, programming and function of individual peripherals.

a. pnx1300 peripheral architecture
b. Data highway bus arbitration
    i. Data highway bus arbitration example
c. Video output peripheral
    i. Video output peripheral configuration
    ii. Video output data format
        1. Video output data sampling schemes
    iii. Video output overlay
        1. Video output overlay configuration
        2. Video output overlay data
        3. Video output overlay data conversion
        4. Video output overlay alpha blending
        5. Video output overlay chroma keying
d. Video input peripheral
    i. Video input peripheral configuration
    ii. Video input data format
    iii. Video input data sampling schemes
e. Image Co-Processor
    i. Image Co-Processor programming
    ii. Image Co-Processor data format
        1. Image Co-Processor input data sampling schemes
        2. Image Co-Processor RGB data format
    iii. Image Co-Processor scaling configuration
        1. Image Co-Processor RGB/YUV conversion
        2. Image Co-Processor overlay configuration
        3. Image Co-Processor overlay data conversion
    iv. Image Co-Processor filter coefficients
f. VLD - Variable Length Decoder
g. Audio output peripheral
h. Audio input peripheral
i. SPDIF digital audio output peripheral
j. pnx1300 PCI / XIO interface
    i. pnx1300 data cache PCI connection
    ii. pnx1300 PCI interface registers
    iii. pnx1300 PCI programming
    iv. pnx1300 XIO interface
    v. pnx1300 XIO examples
k. IIC interface

6. pnx1500 and pnx8550 System on Chip architecture

In-depth look at the pnx1500 System on Chip architecture, and the TriMedia-based peripheral architecture of the pnx8550.

a. pnx1500 peripheral architecture
b. pnx8550 peripheral architecture

c. Video Composition Processor - architecture

i. Video Composition Processor - layers
ii. Video Composition Processor - DMA
iii. Video Composition Processor - data formats
    1. Video Composition Processor - sampling schemes
    2. Video Composition Processor - packed data formats
iv. Video Composition Processor - pool resources
    1. Video Composition Processor - resource assignment
v. Video Composition Processor - pixel formatting
    1. Video Composition Processor - color lookup table
vi. Video Composition Processor - chroma keying
    1. Video Composition Processor - chroma key combining
vii. Video Composition Processor - undithering
viii. Video Composition Processor - upsampling
    1. Video Composition Processor - chroma upsampling
    2. Video Composition Processor - color transient improve
    3. Video Composition Processor - rate conversion
    4. Video Composition Processor - linear interpolator
ix. Video Composition Processor - image improvement
    1. Video Composition Processor - histogram modification
    2. Video Composition Processor - luma sharpening
    3. Video Composition Processor - color features
    4. Video Composition Processor - brightness and contrast
x. Video Composition Processor - color space conversion
xi. Video Composition Processor - layer fetch control
xii. Video Composition Processor - mixer
    1. Video Composition Processor – mixer chroma keying
    2. Video Composition Processor - chroma key combining
    3. Video Composition Processor - mixer keys
    4. Video Composition Processor - raster operations
    5. Video Composition Processor - alpha blending
xiii. Video Composition Processor - output pipeline
    1. Video Composition Processor - chroma downsampling
    2. Video Composition Processor - gamma correction
    3. Video Composition Processor - output format
    4. Video Composition Processor - screen timing
xiv. LCD controller
xv. Video Output Router

d. Video Input Processor - architecture

i. Video Input Processor - test pattern generator
ii. Video Input Processor - video extraction
iii. Video Input Processor - video capture modes
    1. Video Input Processor - planar data formats
    2. Video Input Processor - packed data formats
    3. Video Input Processor - data sampling schemes
iv. Video Input Processor - dithering
v. Video Input Processor - image scaling
    1. Video Input Processor - scaling unit architecture
    2. Video Input Processor - scaling and polyphase filter
    3. Video Input Processor - re-sampling
    4. Video Input Processor - color space conversion
vi. Video Input Processor - storage modes
vii. Video Input Processor - auxiliary data
viii. Video Input Router
ix. Video Input Router - dual video capture

e. Fast Generic Parallel Interfaces
f. pnx1500 PCI / XIO interface
    i. pnx1500 XIO IDE interface
    ii. pnx1500 XIO examples
g. IIC interface
h. VLD - Variable Length Decoder
i. Audio output peripheral
j. Audio input peripheral
k. SPDIF digital audio output peripheral
l. 10/100 Ethernet MAC

m. General Purpose Input / Output

i. General Purpose Input / Output - time stamping
ii. General Purpose Input / Output - sampling
iii. General Purpose Input / Output - IR receiver

n. Memory based scaler

i. Memory Based Scaler - de-interlacing
    1. Memory Based Scaler - Edge Detection de-interlacing
ii. Memory based scaler - scaling
    1. Memory based scaler - scaling and polyphase filter
iii. Memory based scaler - data conversion
    1. Memory based scaler - planar formats
    2. Memory based scaler - packed data formats
    3. Memory based scaler - indexed data formats
    4. Memory based scaler - data sampling schemes
    5. Memory based scaler - color space conversion

o. 2D drawing engine

7. TriMedia CPU Core Architecture

Explanation and clarification of the TriMedia CPU core architecture including its design aims and design choices; instruction format; review of VLIW scheduling; register-based architecture; custom operations; compilation tools; and cache architecture.

a. TriMedia core architecture

b. Superscalar and VLIW scheduling
c. TriMedia core functional units
    i. Functional unit example
d. Instruction format
    i. Instruction set design
    ii. CISC, RISC and Superscalar architectures
e. Data registers
f. SIMD operations overview
g. Compilation tools

h. TriMedia cache architecture

    i. TriMedia data cache mapping
    ii. Cache architectures introduction
    iii. Cache mapping
    iv. Cache set associative mapping
    v. TriMedia data cache banks

i. Simulator
j. Debugger

8. Optimizing TriMedia programs

In-depth treatment of optimization including; optimization strategies; tactics to follow in discovering and exposing possible optimizations; profile-driven optimization; cache optimization; parallel scheduling optimizations; and use of custom operations for optimization.

a. Optimisation
b. Profiler
c. Profile driven optimisation

d. Cache optimization

i. Data cache hit and miss policy
ii. Data cache profiling
iii. Data cache management
iv. Data cache programming
v. Data cache mapping optimisation
vi. Instruction cache
vii. Instruction cache profiling
viii. Instruction cache optimisation

e. Parallel scheduling optimization

i. Scheduling
    1. Functional unit optimisation - schedule report
ii. Decision trees
    1. Grafting
    2. Loop unrolling
    3. Function inlining
iii. Restricted pointers
iv. Optimization example - FIR
v. Register variables
vi. Temporary variables
vii. Software pipelining

f. Mapping the application
g. Tuning the application
h. FIR filter example - background

i. Floating point optimizations

i. Fixed point data types
ii. Fractional fixed point

j. Custom operation optimization

Explanation of custom operations and their use in different approaches to optimization including increased parallelism through Single Instruction Multiple Data (SIMD); reduced memory access overhead through register-based manipulations; SIMD operations for common Media Processing functions; and specific SIMD operations for specialised purposes.

i. SIMD operations overview
    1. SIMD naming convention
    2. SIMD operations - macros

ii. SIMD operation FIR example

    Increased parallelism for classic DSP functions.
    1. SIMD operation FIR example - funshift
    2. SIMD operation FIR example - ifir

iii. SIMD - matrix transposition example

    Optimizing by reducing memory accesses through register-based instead of memory-based data manipulations.
    1. SIMD matrix transposition - using pack and merge

iv. SIMD MPEG kernel example

    SIMD operations for common Media processing functions.
    1. SIMD MPEG kernel example - quadavg
    2. SIMD MPEG kernel example - quadadd
    3. SIMD MPEG kernel example – compacting
       (or, when not to go too far)

v. SIMD median filter example
    1. SIMD median filter example - quadumax and quadumin

vi. SIMD sum of absolute differences example
    1. SIMD sum of absolute differences example - ume