Machine Models
- One Address (6502, PDP-8, CDC-3300)
- Two Address (8088)
- Three Address (ARM, CDC-6000)
- RISC vs. CISC
- Little-Endian vs. Big-Endian
- Register-Memory operations
- Fetch-Execute Cycle
- Program Counter
- RAM Memory Layout, Addressing
- Absolute vs. Page 0 addressing, Indirection
- Subroutine Return Address Styles
  - First Word of Subroutine
  - Pushed on Stack
  - Stored in Register
- Numerical Formats & Conversions
  - Conversions: decimal – hex – binary
  - Naturals (unsigned integers)
  - Integers (signed)
  - Fixed Point
  - Floating Point
  - Types
    - Single Precision
    - Double Precision
    - Extended Precision
  - Components
    - Sign Bit
    - Biased Exponent
    - Mantissa
  - Techniques
    - Adjusting Bias
    - Shifting Mantissa
    - Normalizing Result
  - Concepts
    - +/- Infinite
    - Denormalized Numbers
    - NaN (Not a Number)
    - Zero
- Storage Sizes
  - Byte
  - Word
  - Floating Single
  - Floating Double
- Language Structures
  - While loops
  - Repeat loops
  - If-Then
  - If-Then-Else
- Program Creation
  - Editing Text
  - Assembling
  - Linking
  - Running (in Emulator)
- ARM Registers
  - 16 Integer: R0-R15 (including LR, PC, …)
  - 8 Floating: F0-F7
  - CPSR (NZVC flags)
- Storage Allocation
  - DCD (word)
  - DCFS (single, float)
  - DCFD (double, float)
- Instruction Formats
  - Op Codes
  - Conditional Execution
  - Setting Status Flags
  - Integer Constants (value plus right-rotate)
  - Registers with Shift/Rotate
- Memory Instructions
  - LDR / STR
  - LDFS / STFS
  - LDFD / STFD
- Integer Arithmetic and Test Instructions
  - MOV / MVN
  - ADD / ADC / SUB / SBC / RSB / RSC
  - AND / ORR / EOR / BIC
  - CMP / CMN / TST / TEQ
  - MUL / UMULL
- Floating Point Instructions
  - ADF / SBF / MUF / DVF / RSF / RDF
  - ABS / POW / RPW / SQT
  - CMF / FLT / FIX
  - Constants 0.0 – 5.0, 10.0, 0.5
- Binary Arithmetic
  - Addition
  - Subtraction
  - Shifting Bits
    - Logical Shift
    - Arithmetic Shift
    - Rotating Bits
  - Negating (2’s complement)
  - Masking (AND, OR, EOR)
  - Generation and Use of Carry
  - Detection of Zero
  - Detection of Negative
  - Detection of Overflow
Branches and Conditions
B / BHI / BLS
BEQ / BNE / BMI / BPL
BCC / BCS / BVC / BVS
BGT / BGE / BLT / BLE

Subroutine Issues
Return Address Management
Register Transparency
Passing Parameters
Through Registers
Through Memory
Through Stack
Parameter Passing Types
Call-by-Value
Call-by-Return
Call-by-Value-Return
Call-by-Reference
Allocation of Local Variables
Stack-Frame Management
Use of SP to allocate storage
Use of IP to set stack frame base
Nested Subroutines
Recursion
Use of EQU symbols for stack offsets

Array approaches
Use of Base Registers
Simple Array Indexing/Referencing
Pre-Indexing with Write-back (push)
Post-Indexing (pop)
SP versus IP register usage
1D versus 2D arrays
Non Zero-Based Indexes
Mapping Functions
Use of EQU symbols for array offsets

Combinatorial Circuits
AND / OR / NAND / NOR / XOR / NOT
Half and Full adders
Ripple-carry adder/subtractors
1-of-2^n address Decoders & Selectors
Minimizing Memory Hardware
Relay circuits

Sequential Circuits
Flip-Flops (Set-Reset, D, T, etc.)
Counters & Shift Registers
Static & Dynamic Memory (bits & arrays)
Serial Adders
Relay circuits

Input-Output
Serial vs. Parallel
UARTs
Start Bits
Stop Bits
Bit Timing
Handshaking
Polling vs. Interrupts
Direct vs. DMA vs. Port-based
Interrupt vectors
Maskable vs. non-maskable interrupts
Interrupt Service Routines & latency
Software (SWI) vs. Hardware Interrupts
How is an ISR like/unlike a subroutine?
What information must be saved / restored?

Memory
Registers
L1 and L2 cache
Primary Memory
Cache Mapping functions
Direct Mapping
Associative Mapping
Set-Associative Mapping
Cache Techniques
Write-Through
Write-Back
Cache Replacement Strategies
Oldest Page
Least Recently Used (LRU)

Speed-Ups
Super-Scalar
Pipelining

Miscellaneous Techniques
DeCasteljau Bézier Algorithm
Recursive Fibonacci
Print Number and UDIV10
Self-Modifying Code