Memory-Processing gap

* Treatment of embedded software domain
  + Mono vs multi-programming
    - Tanenbaum ch 3.1, 3.2 3.3 (virtual memory)
    - Monoprogrammed systems (legacy and embedded)
  + RTOS – Scheduling & Concurrency, Virtualisation
    - Burns & Wellings
    - ?
  + Sensors and IO
    - ?
  + Software structure, volumetrics and impact on hardware utilisation/throughput
    - ?
  + Power requirements
    - ?
* Hardware requirements & (and unnecessary features)
  + MMU - TLB, Page Tables (memory resident or no)
    - Tanenbaum 3.3
  + Caches
    - Patterson & Hennessy
  + Memory packages, Secondary Storage (flash pros/cons)
    - LWN.NET, Patterson & Hennessy, Hartley-Jones
  + Hardware/Software Partitioning and Co-processors
  + DMA – memory block transfer
    - Tanenbaum 5.1.4
    - Osdev
    - LDD
* Existing & Historical Implementations
  + Segmentation vs Paging
    - tanenbaum
  + Embedded vs Commodity systems
  + Concrete MMU examples (ARM, x86, and something dead old)
    - Arm.com (http://wiki.osdev.org/ARM\_Overview)
    - ARM SoC arch - S. Furber
  + Kernel Design, examples of mem-mgmt in OSes
    - Extended and Expanded Memory (including PAE)
    - Windows Embedded Compact - Microsoft.com
    - Linaro / Embedded Linux - LDD - something more lightweight!
    - FreeRTOS - freertos.org
* Platform Description
  + Spartan 3 FPGA
    - Internal architecture
  + Memory resources
    - Location, size, technology
  + Picoblaze
    - Comparision to other softcores
    - Desirable features
    - Limitations
* Current Research will orient fundamental design decisions, and opportunity for expansion
  + Composable DynMem (i.e. shared pages)
  + Improved Predictability and latency using Scratchpad mem (optimal caching behaviour implemented in software)
  + Dynamic MMU for RT SoCs

**Memory Hierarchy Gains**

* Dynamic Memory Allocation
  + Virtual address spaces, ABIs
  + Programmer benefits and risks (leaks)
  + Impact on predictability, use of dynamic allocation methods in embedded space
* Multiprogramming, Application Sandboxing/Safety, Shared Memory, Libraries
  + Shared libraries not strictly shared in memory (i.e. duplicate pages)
* Virtual Memory
  + Size vs Speed
  + Paging
  + Improvements in memory latency and bandwidth

Memory Hierarchy Gains/Advantages

* Dynamic Memory Allocation, huge working memory
* Caches can improve memory latency

Cons

* Loss of predictability/determinism
* Complex Logic and power requirements
* Software Overheads