MIRABILIS design

Implement Imaginations

System Level Modeling: Organizing Tasks

05/19/08

Identify Tasks

- Identify List of Tasks
  - Estimate Time (First Order), or Cycles (More Detailed)
  - SubTasks: Processor, Routing (bus, switch), Memory
- Order List of Tasks
  - Most Processing Intensive First
  - Note dependent, independent
- Estimate Rate of Tasks
  - Aperiodic Tasks (packet related)
  - Periodic Tasks (background processes)

Consider Task Execution

- Aperiodic/Periodic Tasks on N Cores?
  - Treats set of Tasks as symmetrical to be distributed to a hardware core based on loading, such as AT1, AT2, AT3... to Core 1.
- Split Aperiodic/Periodic Tasks?
  - Send Aperiodic Tasks to N - M Cores, Periodic Tasks to M Cores.
    - AT1, AT2, AT3 to Core 1, 2, 3
    - PT1, PT2, PT3 to Core 4
- Combinations thereof

80 / 20 Nature of Tasks

- 20 Percent of Tasks can take 80 Percent of Processing
  - Tasks may not be symmetrical in terms of time or resources required.
  - Try to Identify these Tasks to improve overall performance
    - Break them up into smaller subtasks?
    - Allocate these tasks with higher priority?
    - Distribute these tasks, if somewhat independent, to additional Cores?

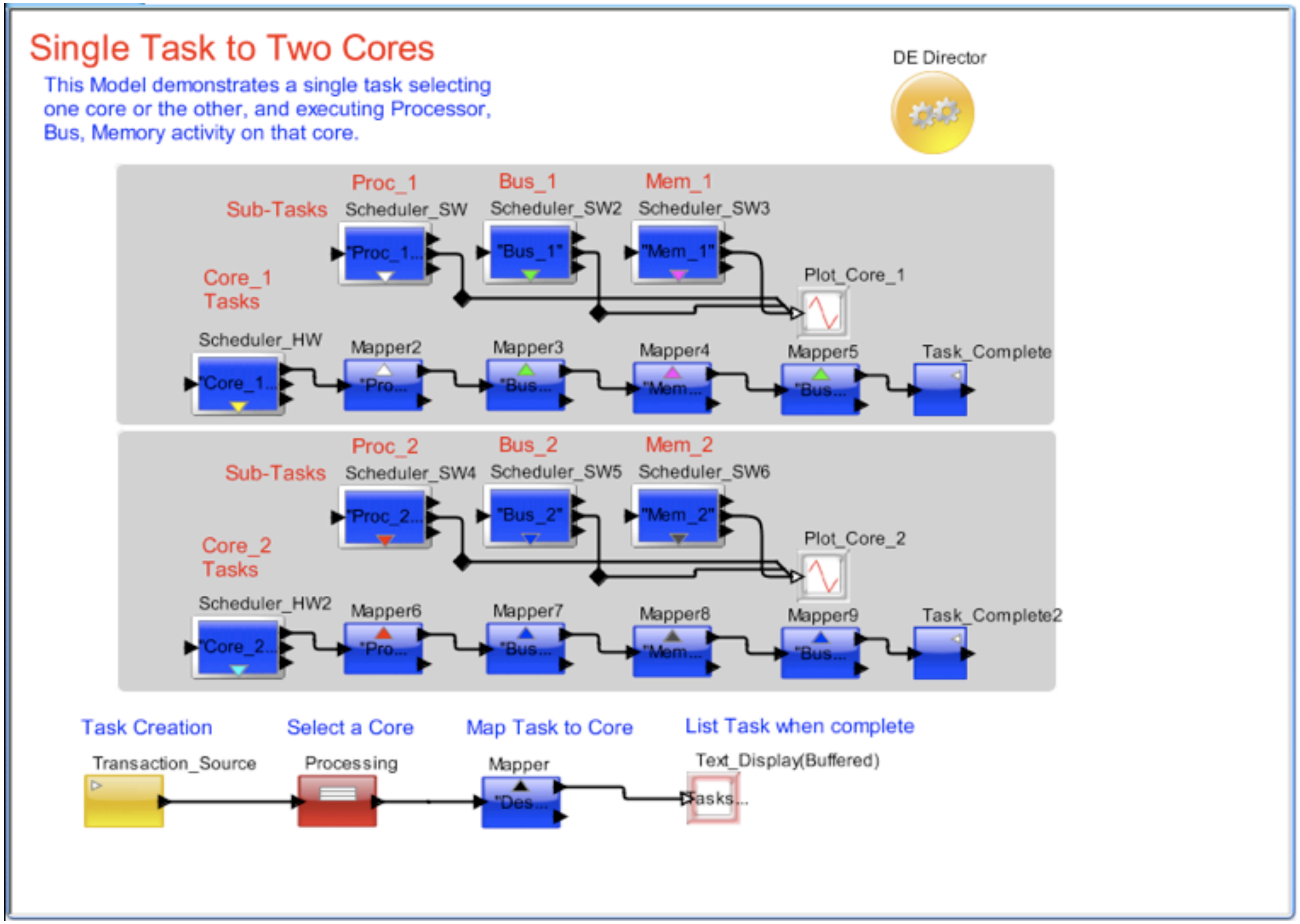
Task Data Structures

- Define Task in a Single Data Structure
  - Common Fields
    - Task Name, Task ID, Time Stamp, Source, Destination, Size, Time to next Task, etc.
  - Sub-Task Fields
    - Ordered List of Resources to use.
    - Ordered List of Cycles, based on Processor, Routing, Memory percentages.
  - Reference Field
    - Model Database for Source, Destination, Hop

Task Flow Considerations

- Conditions may alter Task Flow
  - Different Task Mapping
    - RTOS level
    - Fixed or Variable Processor Core Assignment
  - Different Sub-Task Mapping
    - Hardware level
    - e.g. Cache Miss to SRAM for data (Sub-Task)
  - Behavior-Driven
    - Software level
    - Packet may/may not need certain processing

Single Task to Two Cores



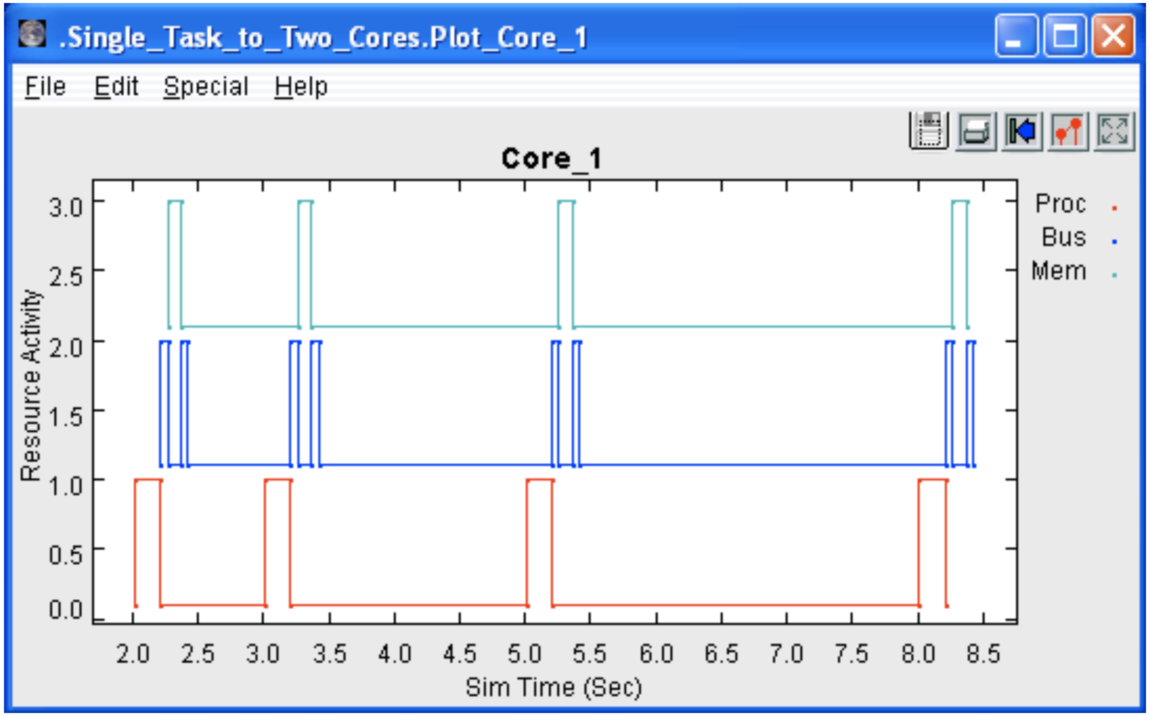
Single Task to Two Cores

This Model demonstrates a single task selecting one core or the other, and executing Processor, Bus, Memory activity on that core.

Task DS Processing

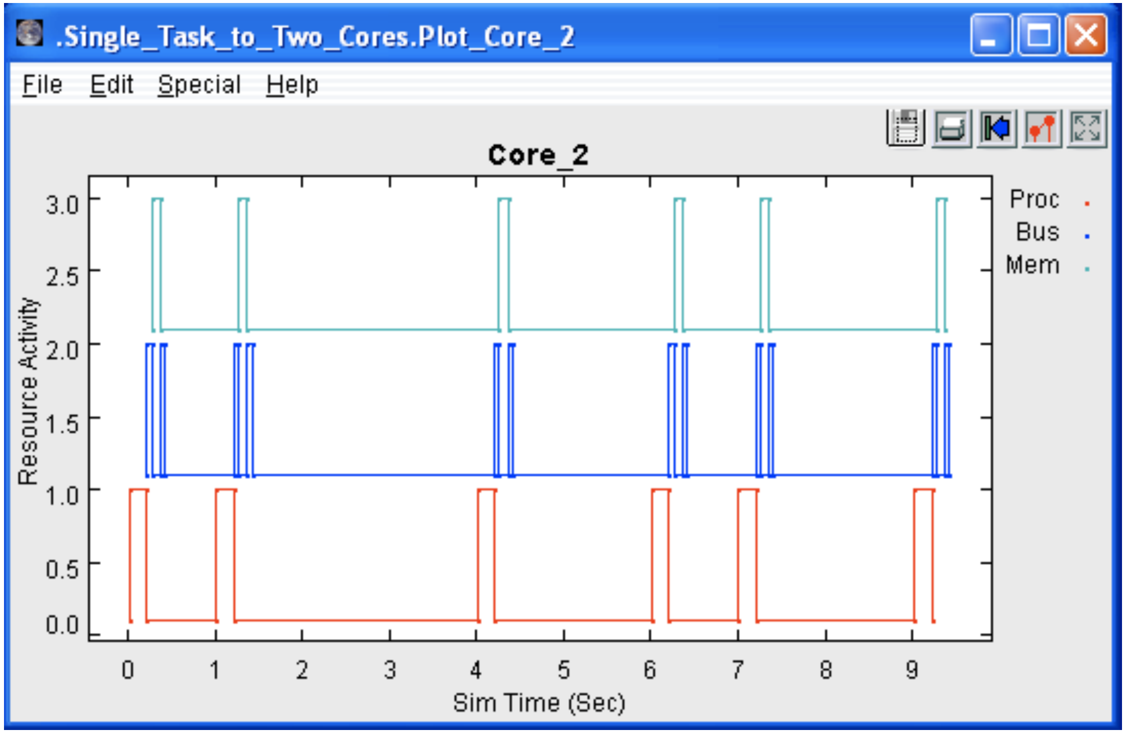
```
input.Task_Name = "Task_1"
input.Task_ID = input.ID
input.Task_Priority = 0
input.Task_Size = 20
input.Task_Time = 0.01
input.Source = "Rx_1"
input.Destination = random() < 0.50 ? "Core_1" : "Core_2"
input.Processor_Cycles = cast(double, input.Task_Size)
input.Bus_Cycles = cast(double, input.Task_Size / 4)
input.Mem_Cycles = cast(double, input.Task_Size / 2)
```

Core_1 Results



Note: Gaps represent Core_2 Task Execution

Core_2 Results



Note: Gaps represent Core_1 Task Execution

Summary

- Identify, Organize Tasks
  - Task Name, Task ID, Time Stamp, Source, Destination, Size, Time to next Task, etc.
- Map Tasks to Cores
  - Receive Sequence, Transmit Sequence to one Core, Different Cores?
  - How to Process large Tasks?
- Test Basic Tasks, add logic intensive tasks
  - Refine model, look at output metrics