Project 3

Kyle Dunn

11//16/2024

CSAS3111BA

Overview

Project 1

A computer System simulator that has key components of a computer system simulates the CPU, Memory, Interrupt handling, and direct Memory Access or DMA. The CPU fetches decodes and executes instructions from memory. CPU checks for data and then access Ram, the use of interrupt handling helps with efficiency.

Project 2

Continuing from the work done in project 2 its key features are process control, memory management, multithreading and system interruptions, instruction cycle, and process control blocks. Project 2 uses L1, and L2 Caches and process management uses scheduling and context switching. It uses multithreading to allow multiple tasks to be done at once.

Project 1 Overview

Module 1

The instructions are pulled from memory by the CPU then it is decoded to find the specific operations it wants like ADD SUB JUMP and LOAD. The program counter manages the instructions, the accumulator does the math and logic, the instruction register keeps the current instruction and status registers tracks the flags.

Module 2

Reads CPU and Memory to write data to certain locations for the storage of data it is put into Cache L1 and L2 for quicker access. The memory is Ram and the Caches (L1 and L2)

Module 3

CPU in module 3 fetches decodes and executes instructions depending on whether it is math or accessing the memory. For module 3 memory has instructions and data by using registers and memory locations.

Module 4

Module 4 is an interrupter for the CPU and an interrupt flag is used to start an interrupter. Check for interrupt function checks for the interrupt setting it if needed interrupt Handler () function is called this function also handles the interrupt itself.

Module 5

Module 5 is the direct memory access or DMA, transfers data between memory and Input/output without the use of the CPU. The Dma Transfer function does the transfer itself. The initiate DMA function starts this.

Project 2 Overview

Module 1

Module 1 is comprised of CPU operations. It has a program counter, accumulator, instruction register, and zero flag. The operation in module 1 are Add subtract Multiply which are math-based operations. Load and Store which handle data from memory and the accumulator.

Module 2

Module 2 is memory management. It has a memory system with two levels of Cache, L1 and L2 respectively and also a memory table to keep track of the memory for the processes. The memory table is created by the field’s Process ID, identify the process memory block, memory start states the starting address of the memory block, memory ends; ends the process at the address. Also, Isfree tells if the block is free or allocated. The function Best-Fit finds the smallest block that can fit in the memory request.

Module 3

Module 3 is process management with multiple processes by process scheduling. This is done by the process control block or PCB. The process control block is comprised of Pid, Pc, acc, and process state. Context switching for this project is done with round robin. This method means multiple processes can run at the same time without affecting each other.

Module 4

Module 4 is interrupting handling and dispatching it does this with timerInterrupt () a function that does timer interrupts, i/oInterrupt does the input and output interrupts and systemCallInterrupt () which does the system interrupt

Module 5

Module 5 is multithreading I did this module with pthreads there are two programs to handle CPU operations and memory management. Pthread\_create makes the threads and pthread\_join keeps everything in order to help the program work before exiting. Error handling is covered by pthread\_create and pthread\_join.

Project 3 Overview

Module1

Uses fetch-decode-execute has registers conditions flags, status registers. It handles math and operations in the environment. The use of mutex helps access memory and ensure the threads works correctly.

Module2

To help the CPU work efficiently it uses RAM, and L1, L2 Caches. Memory allocations and deallocations and a memory table. That tracks and manages memory for the multiple processes. This project also simulates hits and misses to help the CPU.

Module3

Module 3 uses a process control block it allows multiple processes to happen at once. The use of round-robin and priority scheduling to use the CPU more efficiently. The process tracks if it is ready, running, blocking with the use context switching.

Module4

In module 4 the main focus is interrupts with an Input output timer, systems calls these interrupts are assigned to important tasks. The module uses mutex locks to manage shared memory for different processes.

Module5

Module five is about threads and multithreads to help with concurrent executions of CPU operations. The threads cover memory management process scheduling and the interruptions. The use of semaphores help maintain the project and share the processing power for efficient functioning.

Module6

Module 6 is about multithreading and synchronization, keeps track of execution time and memory usage.

Running the simulator

The way to set up the simulator is to open the correct file on your computer and run it and observe the output.

Testing and debugging

The Fetch decode and execute system has the program counter and accumulator updated correctly and stops when it is finished.

For memory management the best-fit method makes it certain the blocks are used correctly.

Interrupt handling to check that the interrupts have the correct parameters to check the data and CPU to make sure they function correctly.

The code keeps track of execution time and tests edge cases and handles divisions by zero.

Conclusion

The aim of this project is to simulate a CPU system, it has a realistic processor with multitasking, inter-processing communication, memory hierarchy and concurrency management. The project manages CPU resources and system tasks and memory handling to have safe access to shared resources.

Screenshots

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated