Common bus system simulation

In this project we are going to perform simulation on 16 bit common bus. To Understand what is common bus let us first discuss what is bus itself, A bus is set of parallel lines that information (data, addresses, instructions and other information)passes on internal architecture of a computer. Information travels on buses as a series of pulses , each pulse representing a one bit or a zero bit Buses are coming in various sizes such as 4 bits, 8 bits, 16 bits, 12 bits, 24 bits, 32 bits , 64 bits, 80 bits, 96 bits and 128 bits.

From the size of bus we can determine that how many bit a bus will carry in parallel. The speed of the is how fast it moves data along the path. This is usually measured in MegaHertz(MHz) or millions of times or second.

Data Carried by bus in a second is called as capacity of the bus. In buses there is concept of internal and external buses, Bus inside a processor is called is called as internal and outer to processor is called as external bus.

A bus master is a combination if circuits , control microchips, and internal software that control the movement of information between major componenets inside the computer.

A processor bus is a bus inside the processor. Some processor designs simplify the internal structure by having one or two processor buses. In a single processor bus system, all information is carried around inside the processor on one processor bus. In a dual processor bus system, there is a source bus dedicated to moving source data and a destination bus dedicated to moving results. An alternative approach is to have a lot of small buses that connect various units inside the processor. While this design is more complex, it also has the potential of being faster, especially if there are

multiple units within the processor that can perform work simultaneously (a form of parallel processing).

Asystem busconnects the main processor with its primary support components, in particular connecting the processor to its memory. Depending on the computer, a system bus may also have other major components connected.

Adata buscarries data. Most processors have internal data buses that carry information inside the processor and external data buses that carry information back and forth between the processor and memory.

Anaddress buscarries address information. In most processors, memory is connected to the processor with separate address and data buses. The processor places the requested address in memory on the address bus for memory or the memory controller (if there is more than one chip or bank of memory, there will be a memory controller that controls the banks of memory for the processor). If the processor is writing data to memory, then it will assert a write signal and place the data on the data bus for transfer to memory. If the processor is reading data from memory, then it will assert a read signal and wait for data from memory to arrive on the data bus.

In some small processors the data bus and address bus will be combined into a single bus. This is calledmultiplexing. Special signals indicate whether the multiplexed bus is being used for data or address. This is at least twice as slow as separate buses, but greatly reduces the complexity and cost of support circuits, an important factor in the earliest days of computers, in the early days of microprocessors, and for small embedded processors (such as

in a microwave oven, where speed is unimportant, but cost is a major factor).

Aninstruction busis a specialized data bus for fetching instructions from memory. The very first computers had separate storage areas for data and programs (instructions). John Von Neumann introduced the von Neumann architecture, which combined both data and instructions into a single memory, simplifying computer architecture. The difference between data and instructions was a matter of interpretation. In the 1970s, some processors implemented hardware systems for dynamically mapping which parts of memory were for code (instructions) and which parts were for data, along with hardware to insure that data was never interpretted as code and that code was never interpretted as data. This isolation of code and data helped prevent crashes or other problems from “ runaway code” that started wiping out other programs by incorrectly writing data over code (either from the same program or worse from some other user’s software). In more recent innovation, super computers and other powerful processors added separate buses for fetching data and instructions. This speeds up the processor by allowing the processor to fetch the next instruction (or group of instructions) at the same time that it is reading or writing data from the current or preceding instruction.

Amemory busis a bus that connects a processor to memory or connects a processor to a memory controller or connects a memory controller to a memory bank or memory chip.

Acache busis a bus that connects a processor to its internal (L1 or Level 1) or external (L2 or Level 2) memory cache or caches.

AnI/O bus(for input/output) is a bus that connects a processor to its support devices (such as internal hard drives, external media, expansion slots, or peripheral ports). Typically the connection is to controllers rather than directly to devices.

Agraphics busis a bus that connects a processor to a graphics controller or graphics port.

Alocal busis a bus for items closely connected to the processor that can run at or near the same speed as the processor itself.

ACCUMULATER: The accumulator processor register in the common bus system is processing unit that help to perform manipulations.

It has two another register Called

1. ADDER AND LOGIC UNIT
2. E REGISTER

ADDER AND LOGIC UNIT: It perform additions and other operation then store the value in the Accumulator.

E REGISTER: It contains the carry of addition and other operation performed in the adder and logic unit.

DATA REGISTER: When we fetched instruction from memory then it is neccesary to have data on which instruction is to be executed.

Data register provide data to instruction to execute it.

TEMPORARY REGISTER: When we are executing instruction then in the way of computing situation arrives when we need a register to save intermediate result.

To save intermediate result we hace register called Temorary register that holds the data or result temporarly from which data will be fetched lator.

INSTRUCTION REGISTER: It tells that which instruction will be ececuted

ADDRESS REGISTER: AR contains the address of the oprends to execute instruction. For example AR(0-11)

PROGRAM COUNTER; It is counter in a common bus that will tell that which instruction will be executed next . Hence it contains the address of next instruction it is implemented as

PC– > PC +1;

INPUT REGISTER: It contains the data that will be inserted by user.

OUTPUT REGISTER: It has data that can be use full to take output.

WORKING OF PROJECT: This project contain one addition display of data which designed with help of graphics function. It is not relate to project at all. But it introduce you what is project.

The main coding when you press any key from key board will appear.

It demands from three control signal s0, s1, s2 these three bits aggregately defines the binary corresponding to which decimal number of the register activated which further give activated the its register and execute instruction In order to display the activated register i have used a pixel and circle that will fill the box.