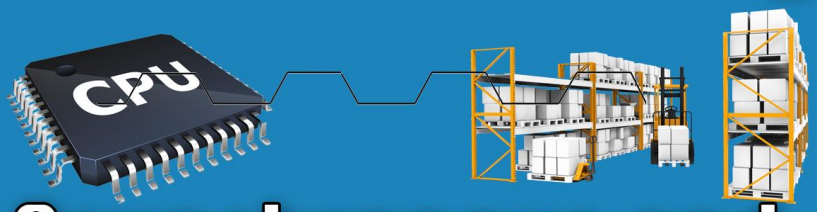
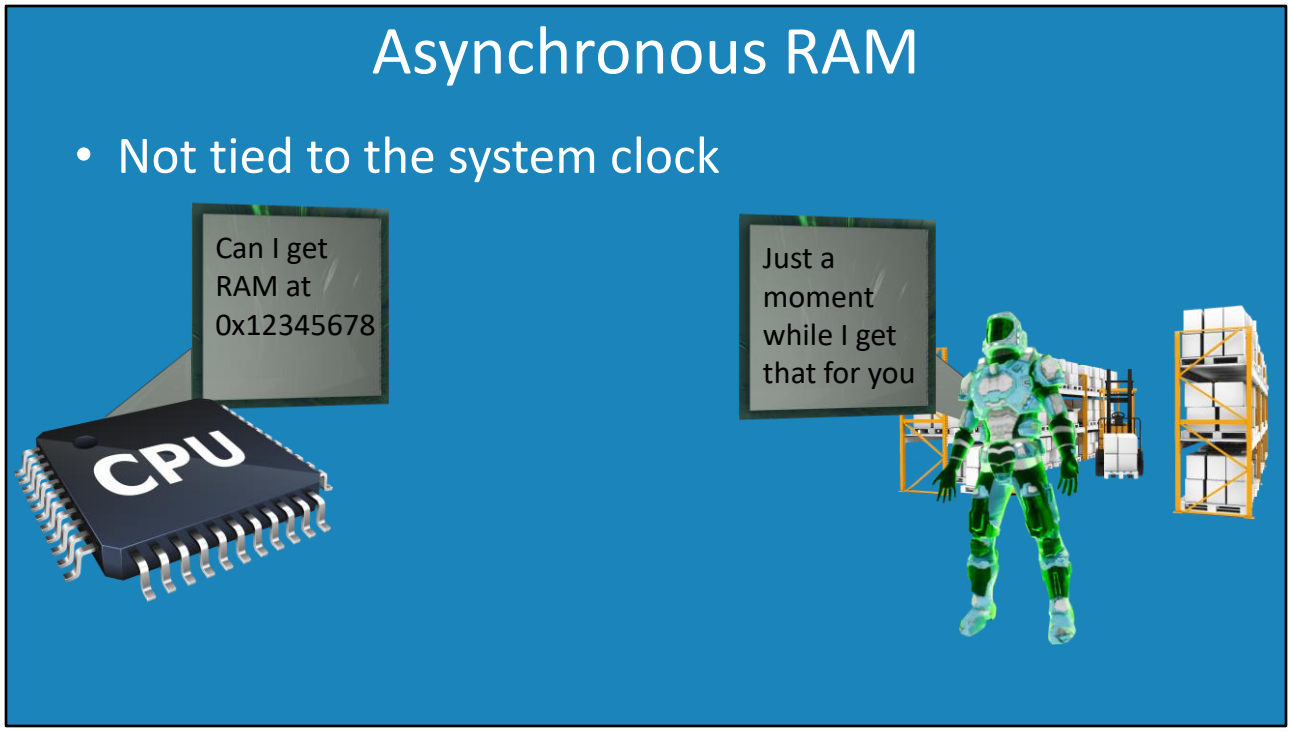
ITFreeTraining



Asynchronous and Synchronous DRAM

For the free video please see
http://itfreetraining.com/ap/3a08

In this video from ITFreeTraining I will look at asynchronous and synchronous memory. Nowadays, computer memory is synchronous, but by looking at both will give you a better understanding of how memory works.

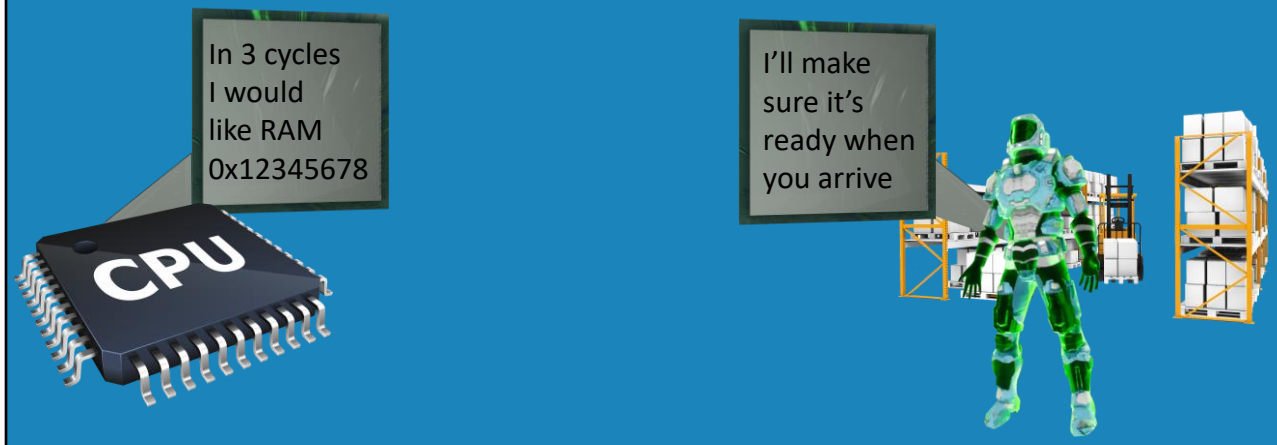


0:0:13 First I will look at asynchronous memory. Asynchronous memory is the older style of DRAM. The newer memory modules are synchronous DRAM which I will cover in a moment. Asynchronous RAM is not tied to the system clock. This means that the operation of the CPU and the memory modules are not synced together. As the CPU and memory modules are not in sync, there may be delays getting data out of the memory modules. This was not a problem in older computers, but as computers started getting faster this became more of an issue.

To understand the process better, consider a CPU that needs access to the contents of a particular RAM location. To do this, it makes a request for the data from the memory modules. Consider that memory is like a warehouse. Once the request is received, someone needs to go and get the item, in this case the contents of that memory address. The request may take time to complete. This causes a delay. The delay could also vary in length. Time that the computer is spending waiting for something is time that could be used to do something else, so is not very efficient.

Synchronous RAM

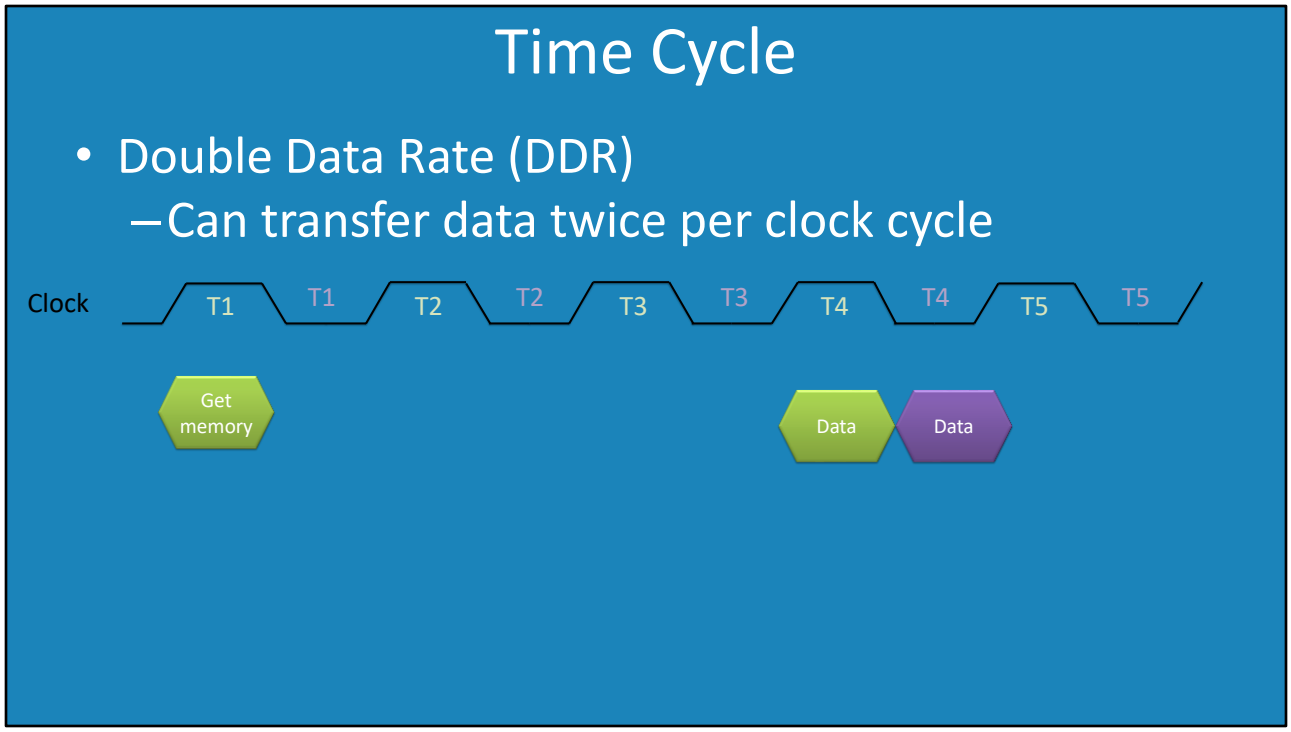
- Tied to the system clock



0:1:23 By contrast, synchronous RAM is tied to the system clock. With synchronous RAM, the CPU will once again ask for the contents of RAM. Unlike before, the CPU will now tell the RAM that it will be back after a certain number of cycles for the contents of the RAM.

If I consider the warehouse scenario again, this time the memory module will retrieve the memory and have it ready at the required time. Using synchronous RAM, the computer knows when the contents of RAM will be ready. The memory module also knows when to have the memory ready. Using this kind of system ensures the computer is not waiting for data to be made ready for it. Everything essentially runs in sync.

I will now look at a different example of synchronous memory as it leads to another important point.



0:2:13 To understand synchronous memory better, consider the following time cycle. The CPU generates the time cycle. The memory module receives the time cycle and synchronizes its operations to this time cycle.

This is a simplified example but will give you the basic idea of how it works. Now consider that a request to get the contents of some memory is made. The memory module will determine how many cycles it will take to retrieve the contents. When purchasing RAM, you may have noticed the tech specification of the RAM is a number of latency values. This is referring to how many clock cycles the RAM will take to perform specific actions. In later videos I will look at these latency values in more detail.

You may also notice that the clock cycle rises and falls. In 1996, there was the release of a new kind of memory module called Double Data Rate or DDR. DDR was very different to previous memory in that it could transfer twice the data given the same clock cycles.

To understand how it can do this, consider the clock cycle has a low and high point. DDR allows the clock cycle to be divided into two. When a single request is sent through, the data can be returned, on both the rise and fall of the clock cycle. Using this method allows DDR to return twice as much data as none DDR RAM given the same number of clock cycles.

This is a simplified example; in reality there are some rules on how memory can be accessed that need to be followed. There are a lot more steps involved than those shown here. In later

videos I will look at the internal working of memory in more detail.

This concludes this video on asynchronous and synchronous memory. I hope that you have found it informative. Until the next video from us, I would like to thank you for watching.

References
"Logic and Computer Design Fundamentals Fourth Edition" pages 433-434, 455
https://en.wikipedia.org/wiki/Dynamic_random-access_memory

Credits
Trainer: Austin Mason http://ITFreeTraining.com
Voice Talent: HP Lewis http://hplewis.com
Quality Assurance: Brett Batson http://www.pbb-proofreading.uk