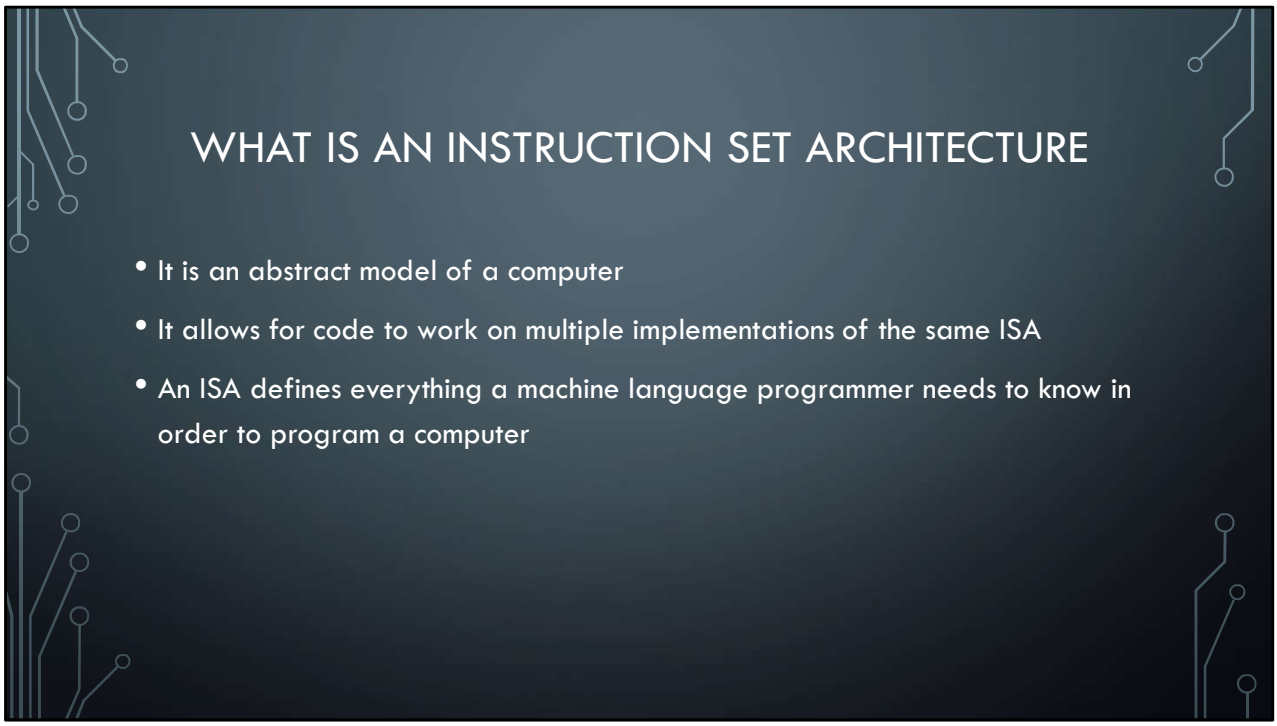

INSTRUCTION SET ARCHITECTURE
ERIC OLIVER

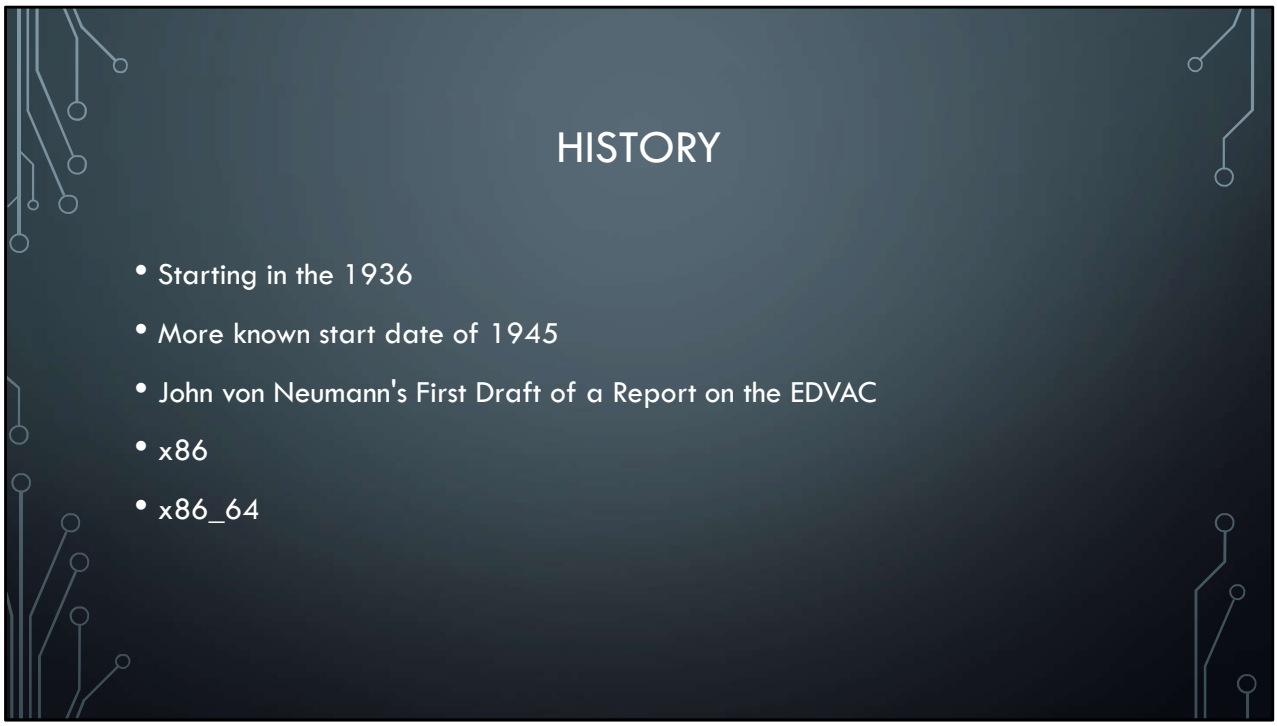
INTRODUCTION

- What is an ISA
- Examples of ISA
- Why They Are important



Have you ever wondered why, your laptop can run the same programs as your desktop; even though they have totally different hardware inside of them. That's because of ISA, specifically the x86_64 architecture, which we will touch on later. The ISA acts as a bridge between the software and the hardware. This has enabled us to diversify into the families of operating systems we have today. And it is also the reason why two machines running windows can, more or less, run the same software.

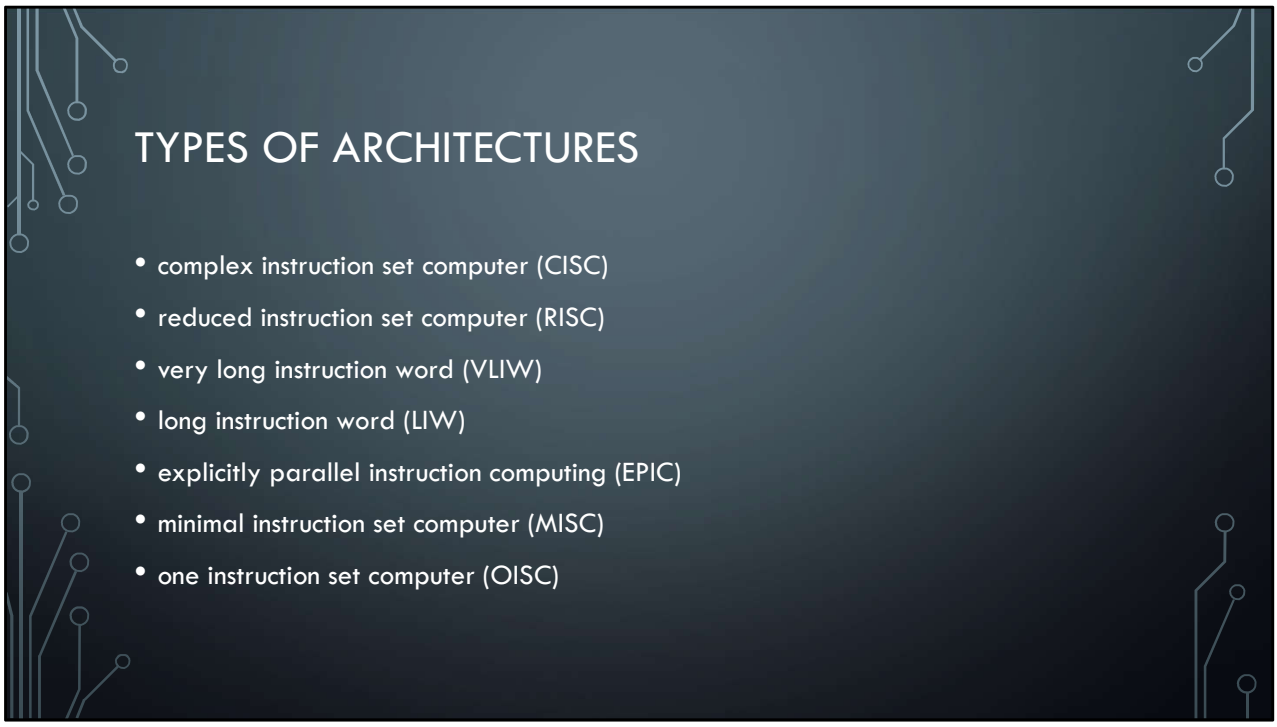
Because we can write code for one machine, it means we can spend a lot less time rewriting the same code over and over again. This also allows us to make cheaper machines and still use the same code we have already written, which lowers the barrier of entry to computing.



Believe it or not the first talks of an instruction set architecture was by Charles Babbage and Ada Lovelace in 1936
Our more common interpretation was in 1945, in a paper written by John Von Neumann.

The current architecture that most systems are running today were started in the early 1960's with the x86 architecture families.

We are still using the x86 architecture, albeit an extension the x86_64 architecture. AMD actually helped Intel make the 64-bit processors more popular. Intel originally wanted to scrap x86 and move totally on to IA-64 architecture but it failed horribly and AMD decided to just extend the x86 architecture and we now have the x86_64

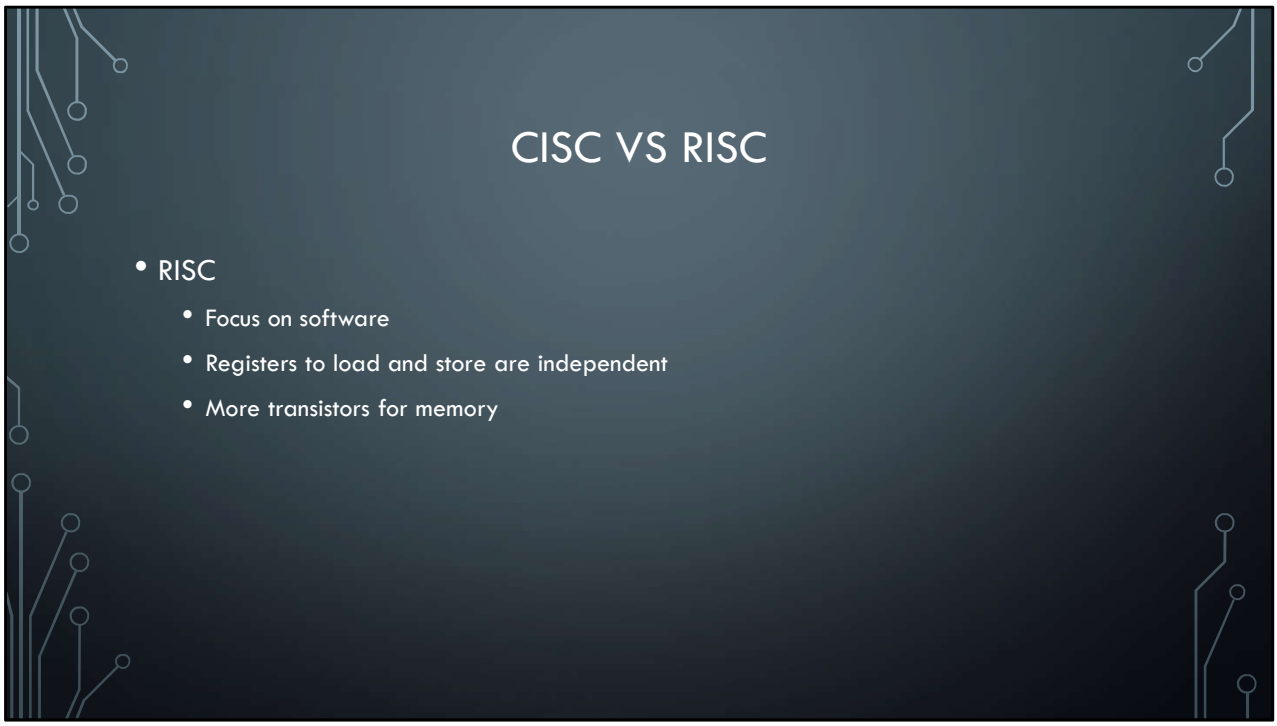


There are many different types of architectures but the main ones we will be talking about today are CISC and RISC.

The other architectures on this list are more experimental with the last two being pretty much theoretical.

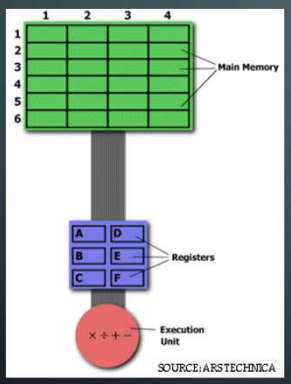
The VLIW, LIW, and the EPIC all seek to exploit parallels in the instructions, by dynamic scheduling of instructions.

CISC architecture focuses on hardware more than it does software. It Uses a greater amount of transistors than RISC boards due, solely because it needs more space to store its more complex instructions.



RISC systems focus more on software control than it does hardware, which means its usually cheaper to make a RISC board than it does a CISC board

EXAMPLE (RISC)

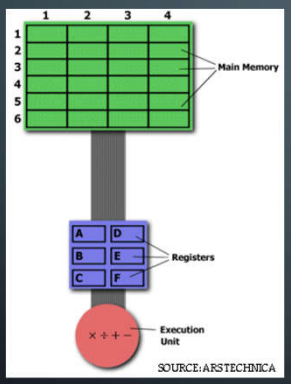


Problem: multiply positions (2,5) by (5,3) and store the result back into 2,5

RISC Approach:

- LOAD A,(2,5)
- LOAD B,(5,3)
- PROD A,B
- STORE (2,5),A

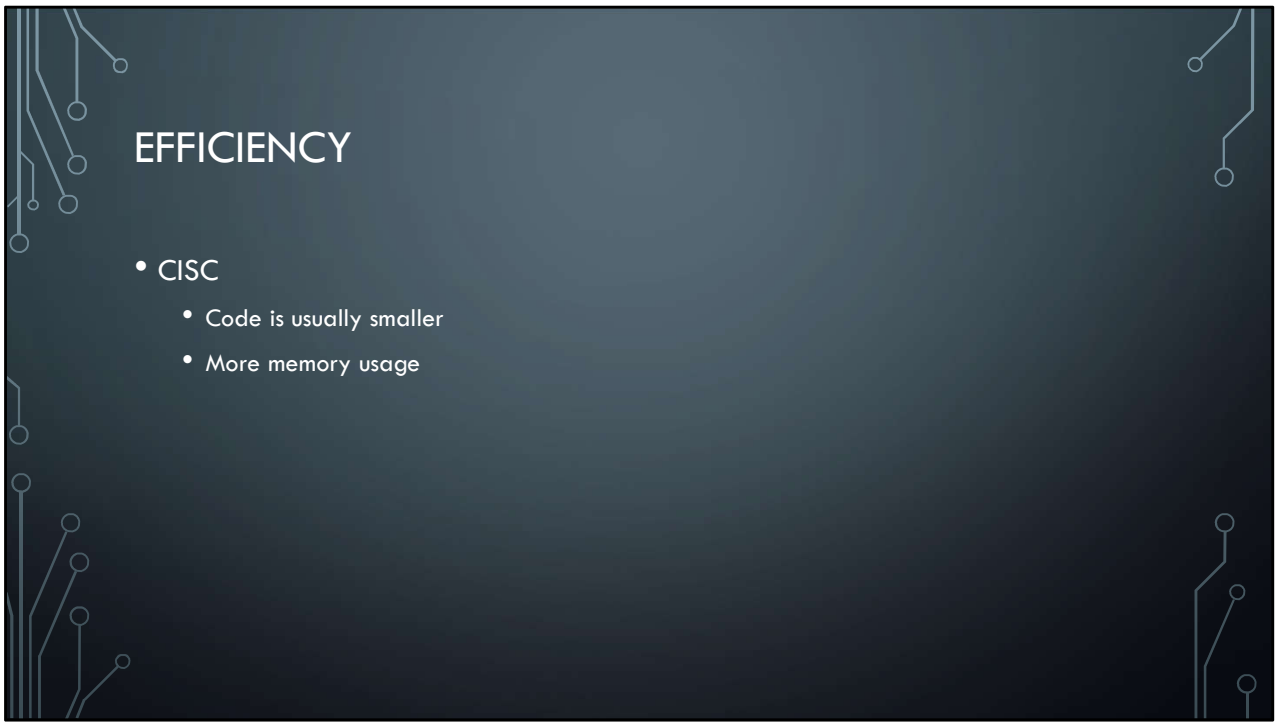
EXAMPLE (CISC)



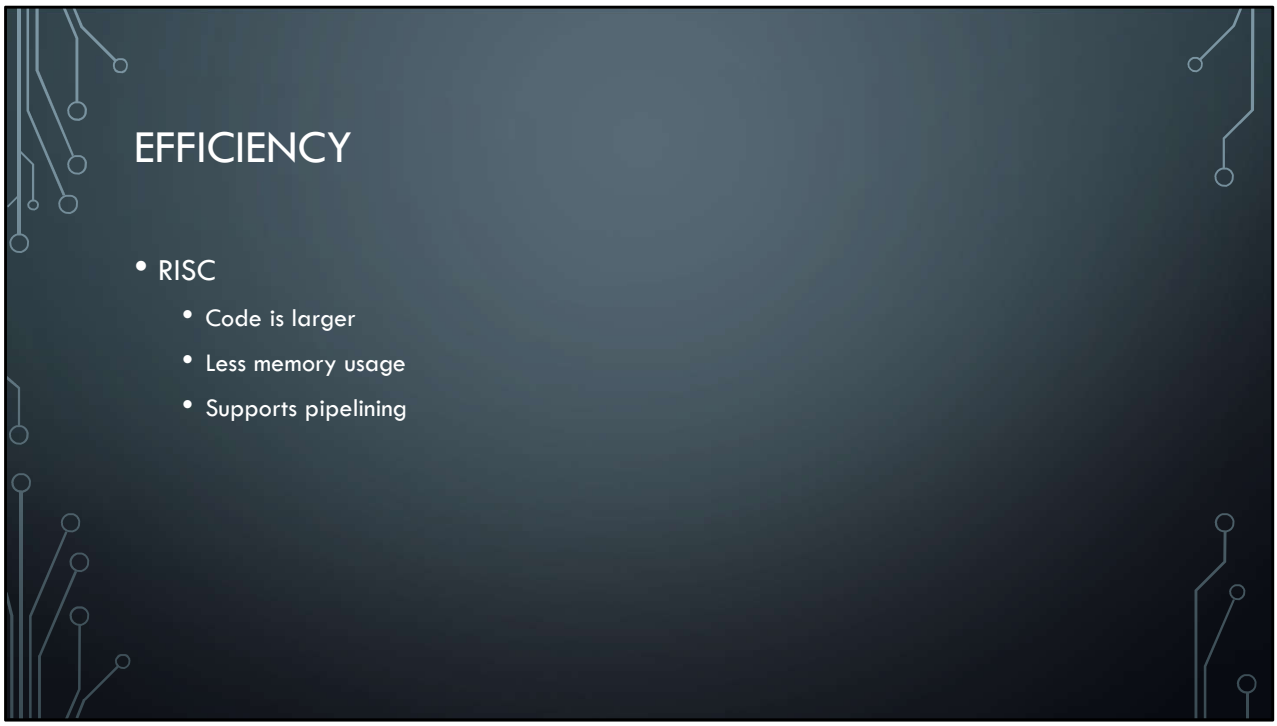
Problem: multiply positions (2,5) by (5,3) and store the result back into 2,5

CISC Approach:

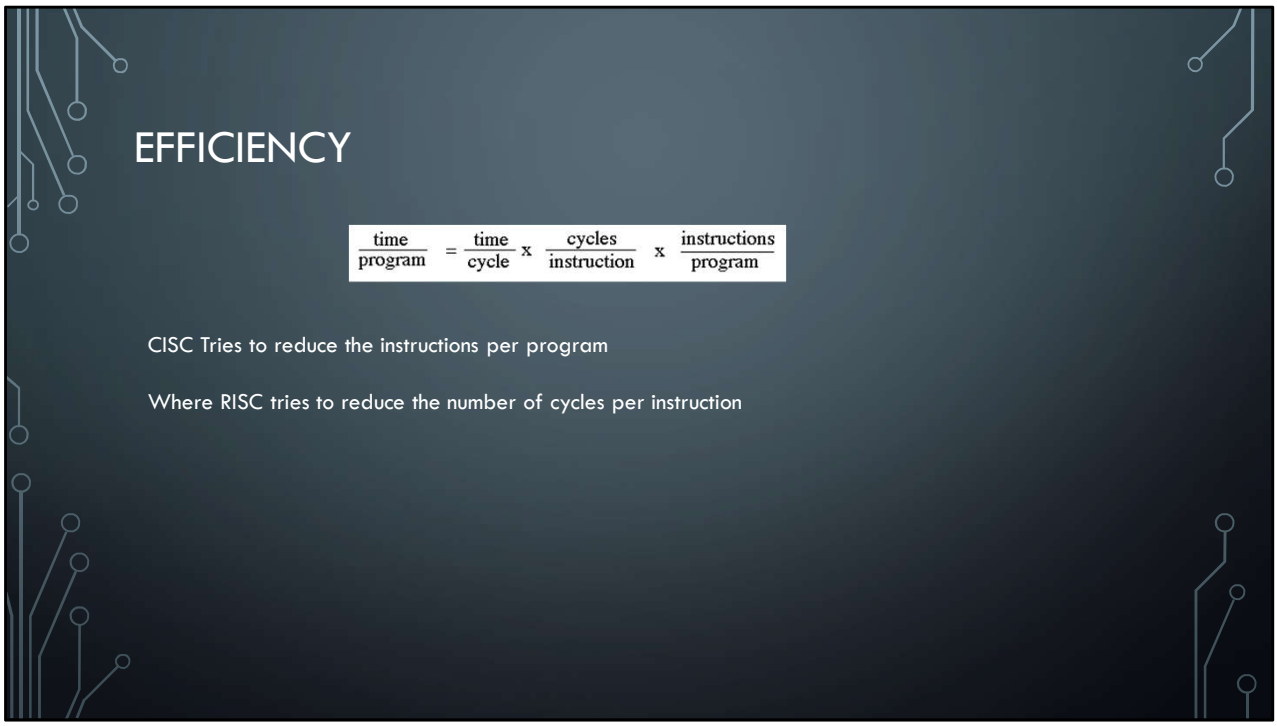
- MULT (2,5),(5,3)



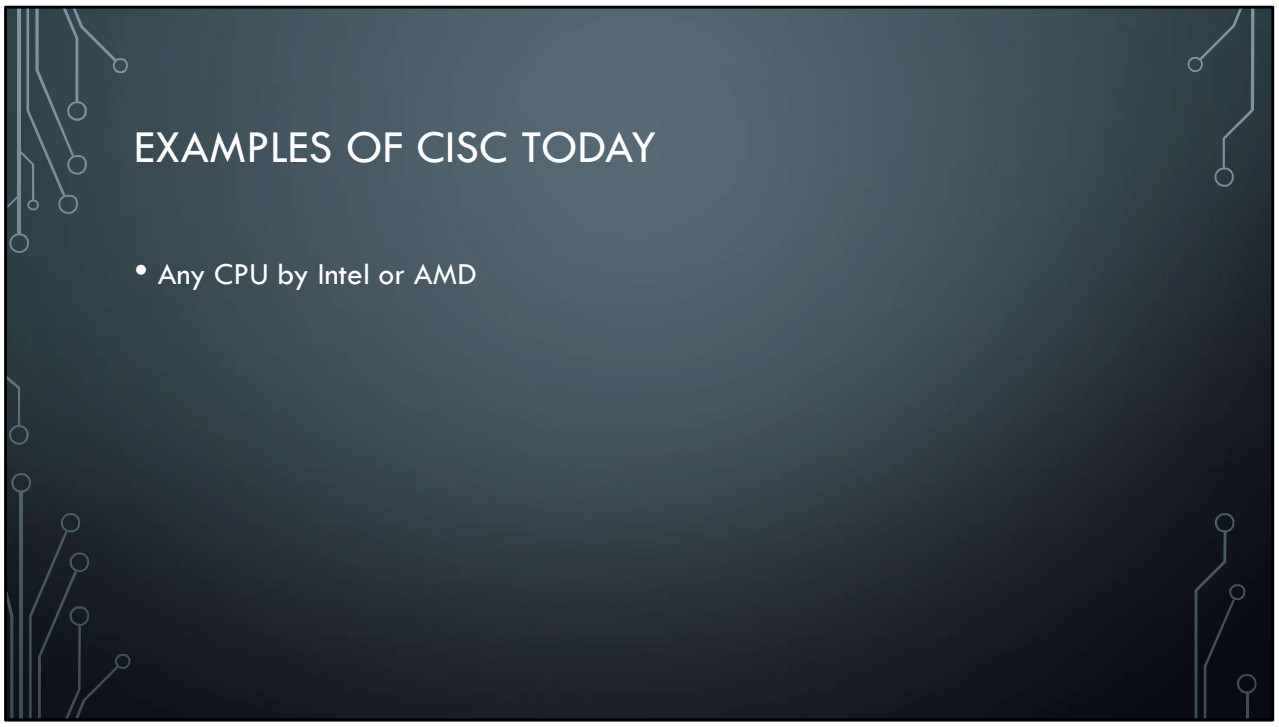
The space to write a program on a CISC machine is usually a lot less than it would be on RISC machine. This means that it does use more hardware costs.



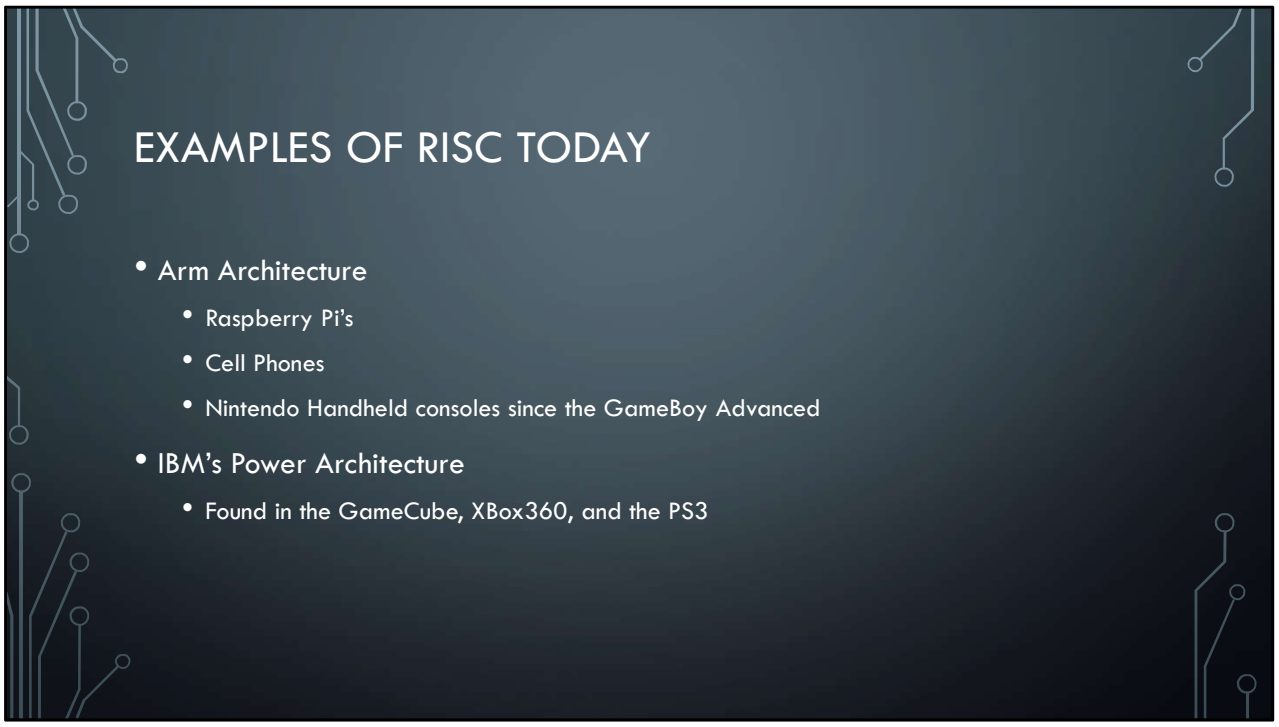
RISC machines do use less memory than CISC machines, mainly because they have less instructions. But since RISC machines do not have as many commands, the code is usally a lot longer. One of the side effects of this is that if you call one of the complex instructions on a CISC machine it does hold the values in registers. Where in a RISC machine it ould store the values in the registers until the are written over, saving the computer work.



Simple Diagram to show the efficiency of a program. CISC machines try to reduce the number of instructions per program, where as the RISC machines tries to reduce the number of cycles per instructions.



Talking about all the pro's and con's makes you think that cisc machines are generally better. Well Here is a list of all the CPU's that use CISC architecture. It is pretty light but it does make up a significant market share.



Conversely Here are some of the many examples of RISC machines today. We have the arm architecture; which runs on the raspberry pi, most major cel phones including iOS and Android. It also runs on every Nintendo system after the Gameboy Andvanced, and yes that does include the switch. IBM's power architecture also runs on RISC architecture which has been in pretty much every game console since the gamecube. Although since the PS4 and Xbox one have switched to a different, but still RISC architecture.

CONCLUSION

- Without ISA's we would not have modern computing
- ISA's allow us to write code on one machine and use it on other implementations of the same ISA

QUESTIONS

- What are the two most common architectures?
  - CISC and RISC
- What Major Company Uses Arm processors in its products?
  - Nintendo
- When Was the earliest architecture talked about?
  - 1936

WORKS CITED

- https://cs.stanford.edu/people/eroberts/courses/soco/projects/risc/risccisc/
- http://archive.arstechnica.com/cpu/4q99/risc-cisc/rvc-5.html#Branch
- https://software.intel.com/en-us/articles/intel-sdm