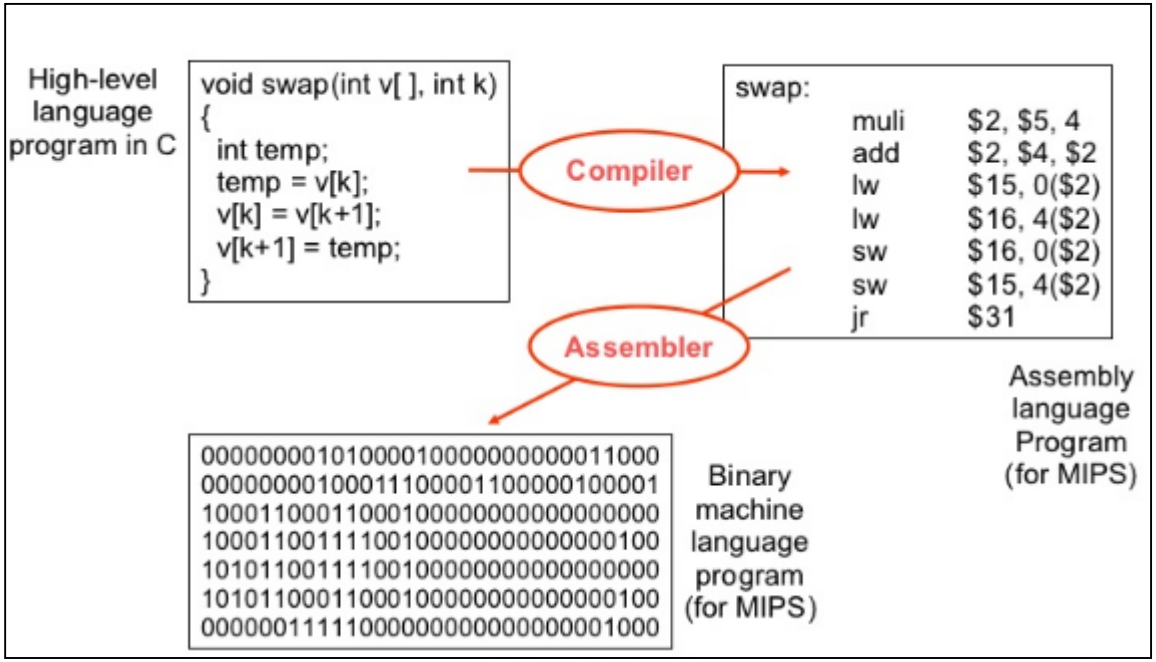
Why is Assembly Language important?
The big picture



Why is Assembly Language important?
Recap: compiling process

- From Embedded Systems I, we learnt
  - A processor only accepts machine code (opcodes & memory/register address).
  - A compiler is used to converting a source file in a .c file extension to an executable binary file in a .elf file extension

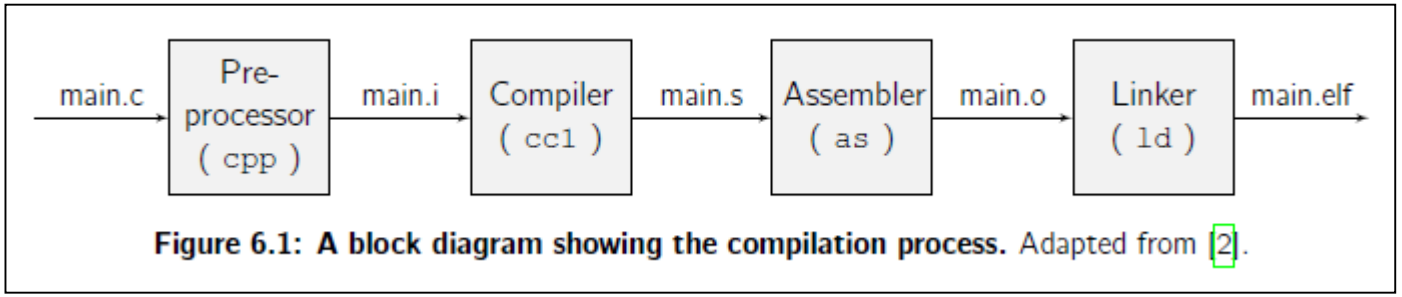


Figure 6.1: A block diagram showing the compilation process. Adapted from [2].

Why is Assembly Language important?
Recap: compiling process

- From Embedded Systems I, we learnt
  - A processor only accepts machine code (opcodes & memory/register address).
  - A compiler is used to converting a source file in a .c file extension to an executable binary file in a .elf file extension
- Compiling process involves
  - Pre-processor: populates pre-processor commands prefaced by the # character.
  - Compiler: converts a C program into an assembly program. The final address for each section has not been assigned as yet.
  - Assembler: converts an assembly program into a binary object file. The final address for each section has not been assigned as yet
  - Linker: links all objects final together into a single executable binary file, with a .elf file extension, where the final address for each section is assigned.

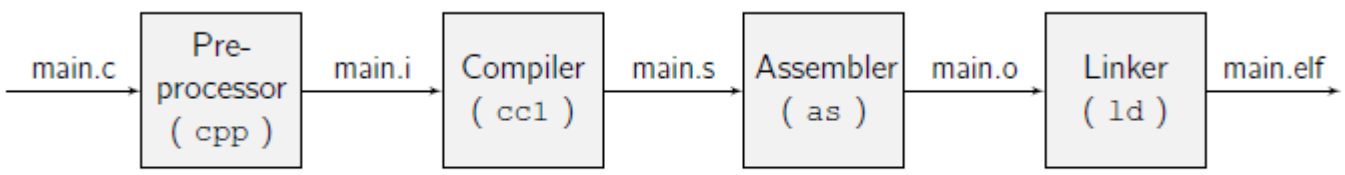


Figure 6.1: A block diagram showing the compilation process. Adapted from [2].

Why is Assembly Language important?
CPU only processes machine instructions

- Let's look a portion of the final .elf file & the related assembly code
- Portion of a program: add four numbers together
  - Total = a + b + c + d , where total, a, b, c, d are all variables stored in memory

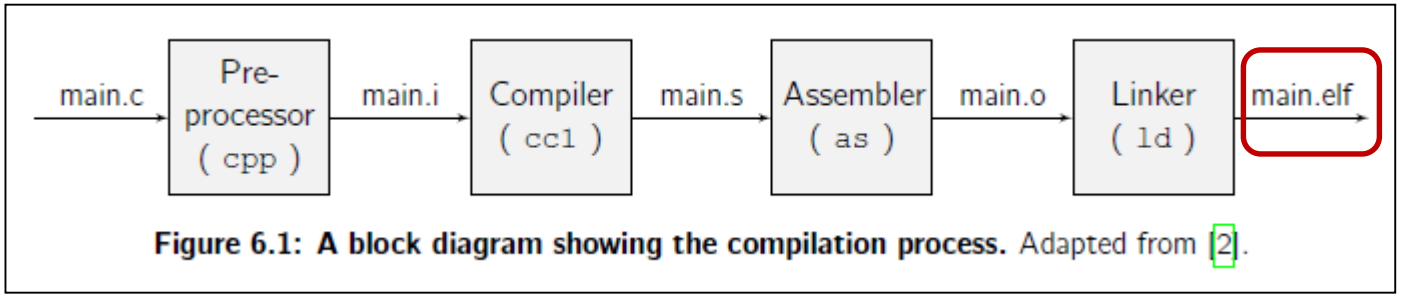


Figure 6.1: A block diagram showing the compilation process. Adapted from [2].

Why is Assembly Language important?
CPU only processes machine instructions

- Let's look a portion of the final .elf file & the related assembly code
- Portion of a program: add four numbers together
  - Total = a + b + c + d , where total, a, b, c, d are all variables stored in memory

…	…	…	…	…
20	0x00010000	MOV	r0, r1	E1A00001
21	0x00010004	ADD	r0, r0, r2	E0800002
22	0x00010008	ADD	r0, r0. r3	E0800003
23	0x0001000C	ADD	r0, r0, r4	E0800004
…	…	…	…	…
line number	address	assembly instructions		machine Instructions (32 bits)

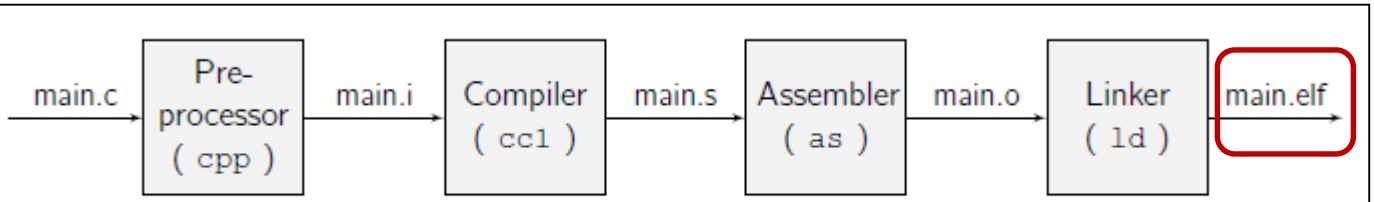
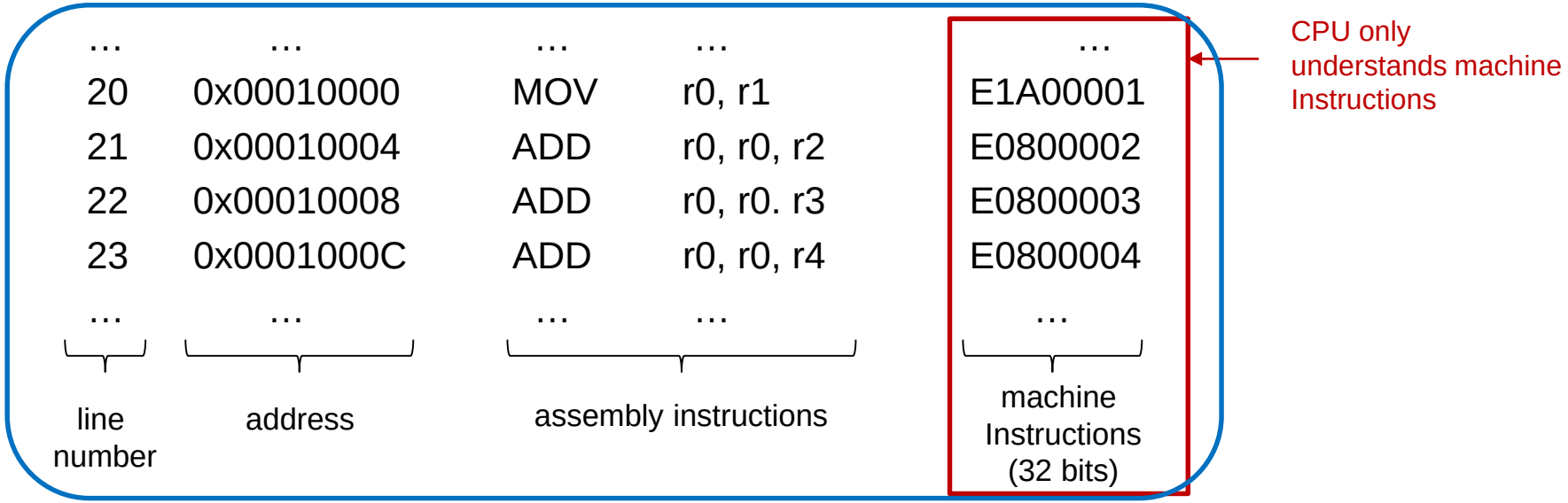


Figure 6.1: A block diagram showing the compilation process. Adapted from [2].

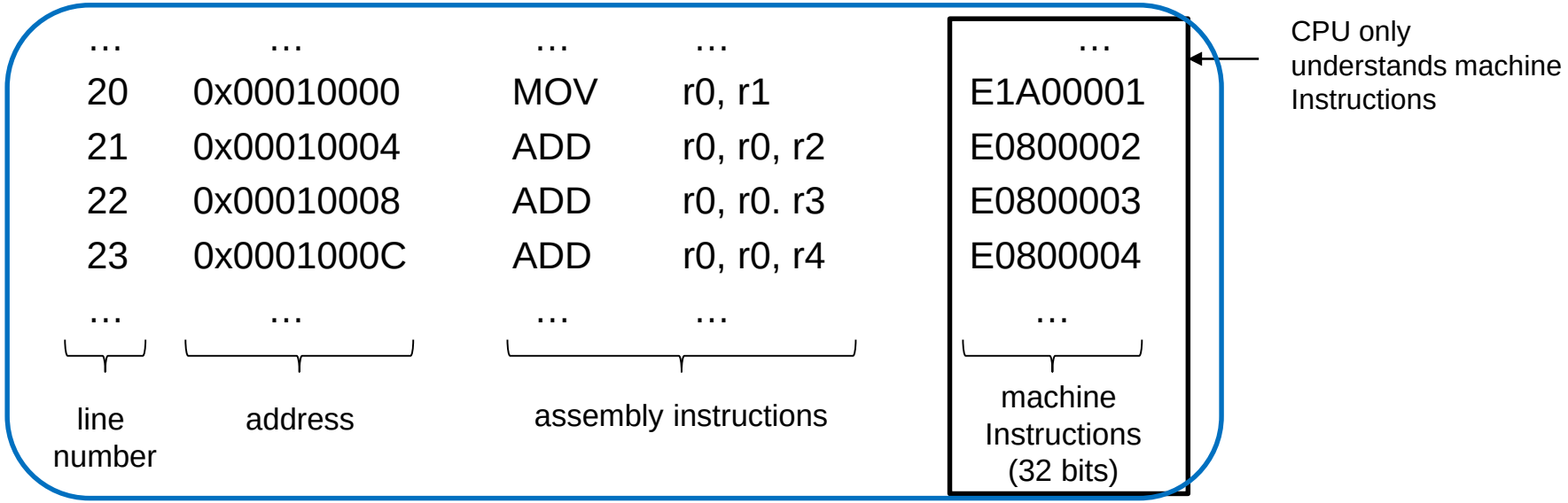
Why is Assembly Language important?
CPU only processes machine instructions

- Let's look a portion of the final .elf file & the related assembly code
- Portion of a program: add four numbers together
  - Total = a + b + c + d , where total, a, b, c, d are all variables stored in memory



Why is Assembly Language important?
CPU only processes machine instructions

- Let's look a portion of the final .elf file & the related assembly code
- Portion of a program: add four numbers together
  - Total = a + b + c + d , where total, a, b, c, d are all variables stored in memory

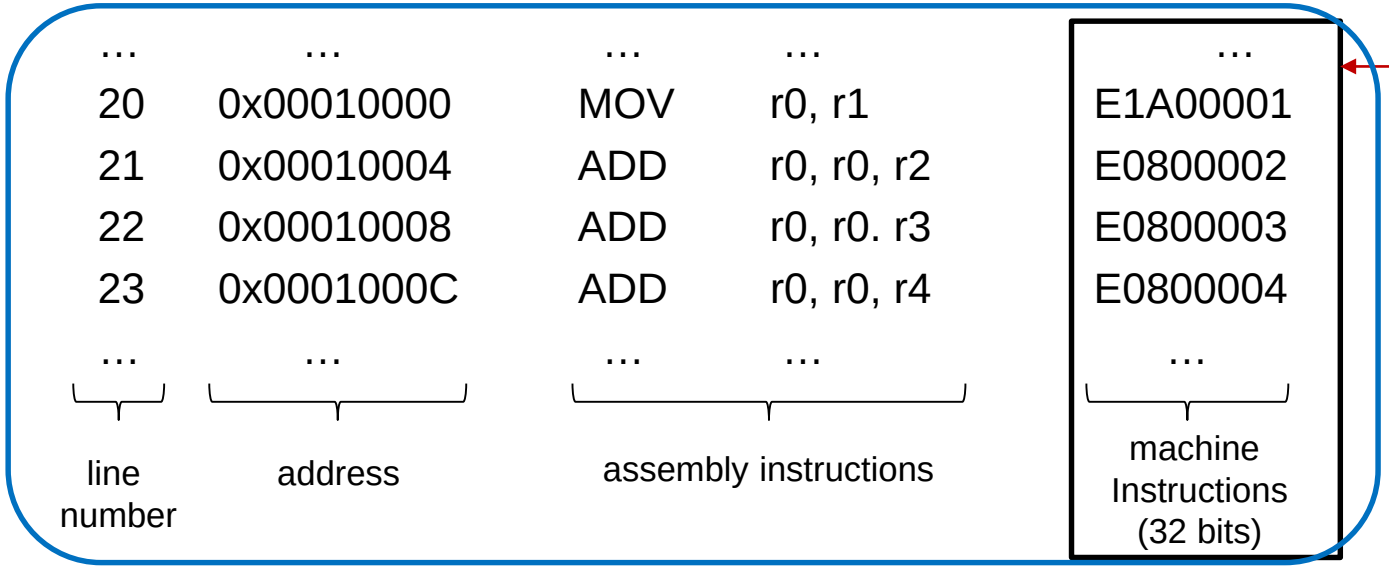


A 32-bit machine instruction is made up of:

- An opcode, which specifies what operation the ALU needs to perform
- Memory/register addresses, which specifies the inputs of the ALU

Why is Assembly Language important?
CPU only processes machine instructions

- Let's look a portion of the final .elf file & the related assembly code
- Portion of a program: add four numbers together
  - Total = a + b + c + d , where total, a, b, c, d are all variables stored in memory



A 32-bit machine instruction is made up of:

- An opcode, which specifies what operation the ALU needs to perform
- Memory/register addresses, which specifies the inputs of the ALU

Why is Assembly Language important?
Compilers

- Compiler setup
  - The compiler needs to be configured carefully for a target processor of choice, since processors have different instruction sets and addressing modes
  - The process of configuring the compiler is known as setting up a toolchain
  - A toolchain is a collection of software tools & programs, which are used to develop source code, convert to a .elf file and debug & run on the target processor

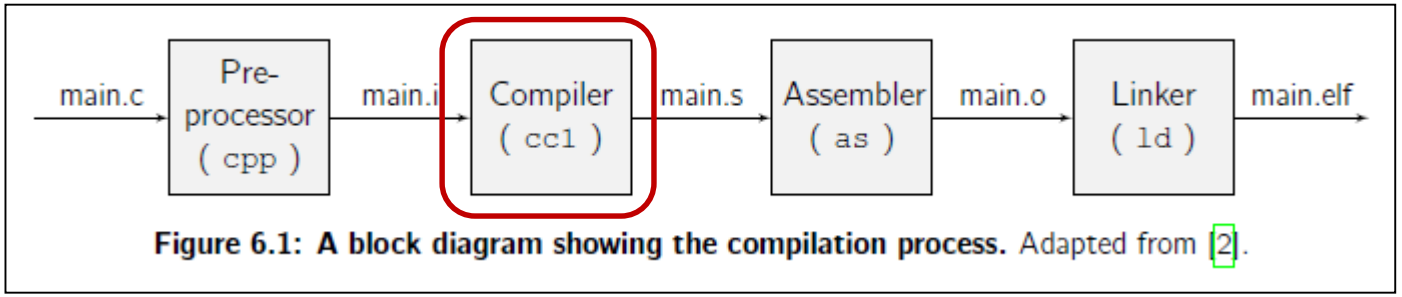


Figure 6.1: A block diagram showing the compilation process. Adapted from [2].

Why is Assembly Language important?
Compilers

- Compiler setup
  - The compiler needs to be configured carefully for a target processor of choice, since processors have different instruction sets and addressing modes
  - The process of configuring the compiler is known as setting up a toolchain
  - A toolchain is a collection of software tools & programs, which are used to develop source code, convert to a .elf file and debug & run on the target processor
- What are the disadvantages of compilers?
  - Compilers do not always generate efficient assembly code for the program to execute as fast as possible.

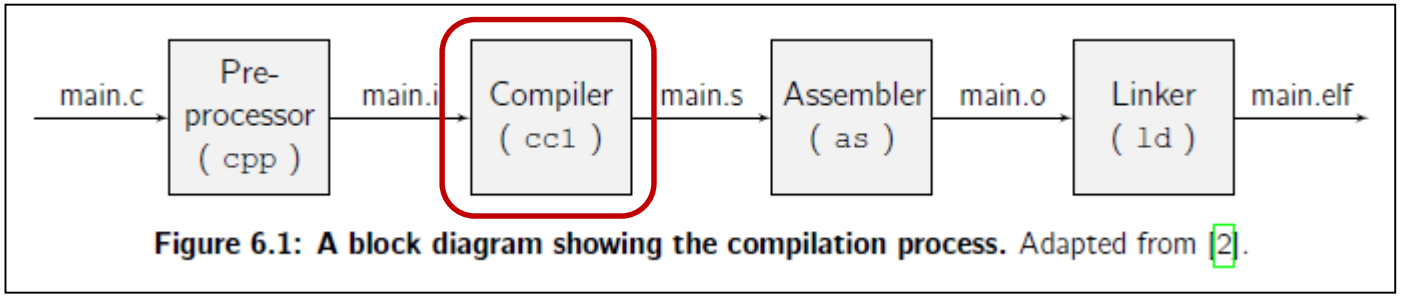


Figure 6.1: A block diagram showing the compilation process. Adapted from [2].

Why is Assembly Language important?
Compilers

- Compiler setup
  - The compiler needs to be configured carefully for a target processor of choice, since processors have different instruction sets and addressing modes
  - The process of configuring the compiler is known as setting up a toolchain
  - A toolchain is a collection of software tools & programs, which are used to develop source code, convert to a .elf file and debug & run on the target processor

- What are the disadvantages of compilers?
  - Compilers do not always generate efficient assembly code for the program to execute as fast as possible.
  - So, what can be done to develop code that will execute as fast as possible?

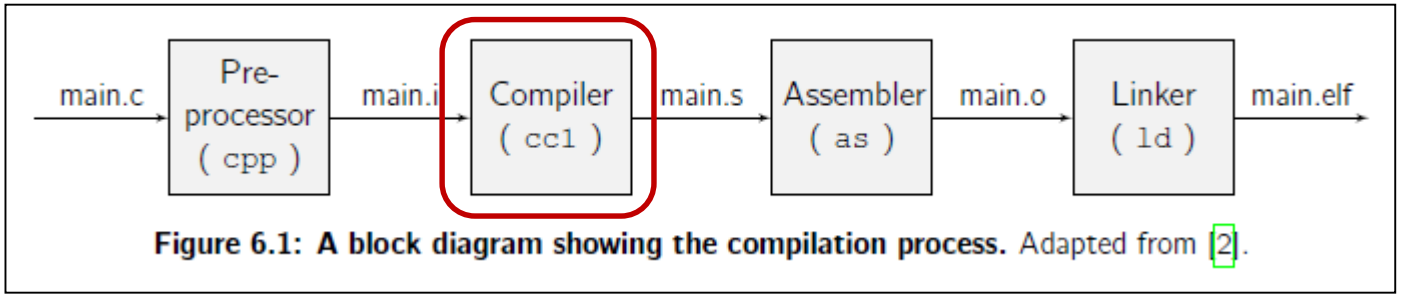
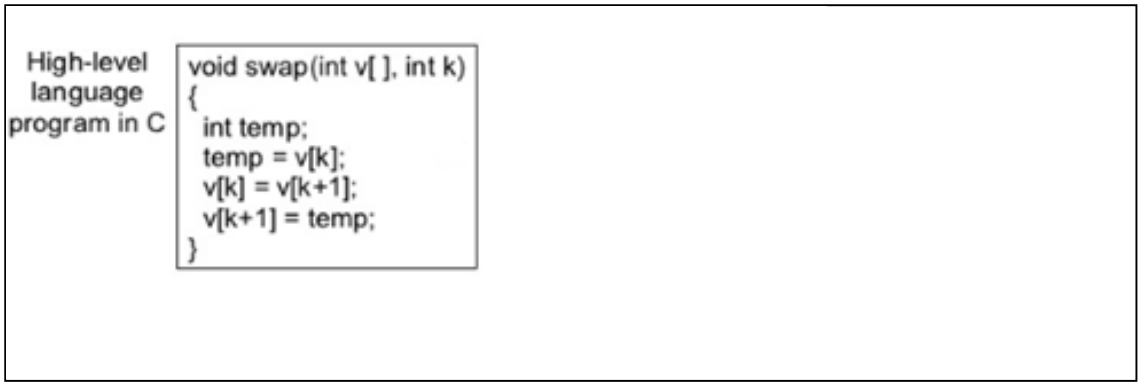


Figure 6.1: A block diagram showing the compilation process. Adapted from [2].

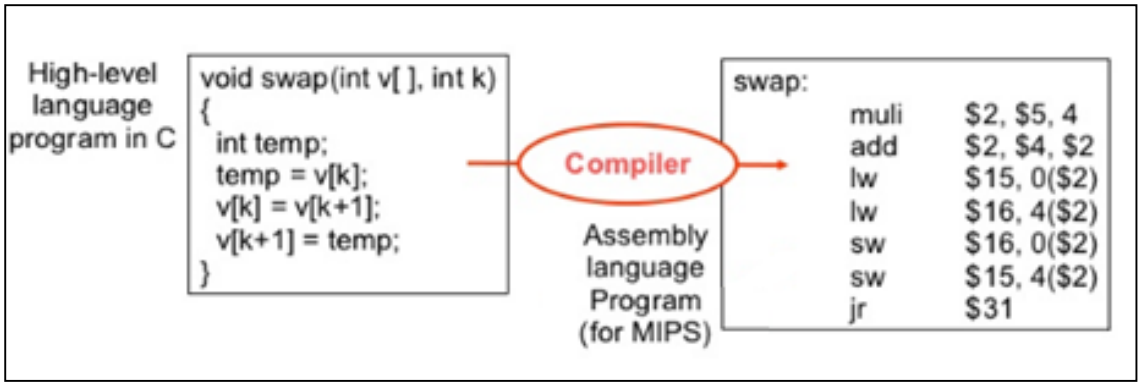
Why is Assembly Language important?
Developing fast and efficient programs

- How can we develop programs that execute as fast as possible?
  - Step 1: developer writes the entire program in a high level language (Examples: C or python)



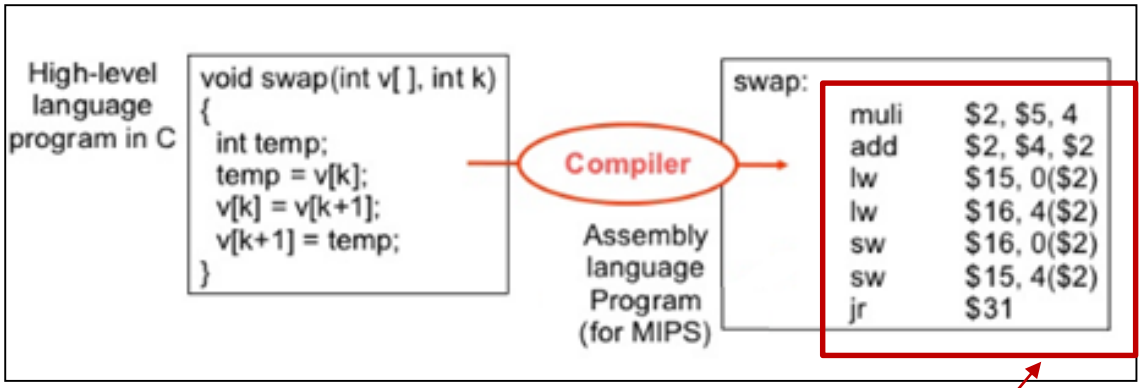
Why is Assembly Language important?
Developing fast and efficient programs

- How can we develop programs that execute as fast as possible?
  - Step 1: developer writes the entire program in a high level language (Examples: C or python)
  - Step 2: identify portions of the code that are ‘timing critical’ that need to be execute as fast as possible
  - Step 3: have the compiler generate assembly code for the ‘timing critical’ portions of code



Why is Assembly Language important?
Developing fast and efficient programs

- How can we develop programs that execute as fast as possible?
  - Step 1: developer writes the entire program in a high level language (Examples: C or python)
  - Step 2: identify portions of the code that are 'timing critical' that need to be execute as fast as possible
  - Step 3: have the compiler generate assembly code for the 'timing critical' portions of code
  - Step 4: the developer can either fine-tune the non-efficient assembly code generated by the compiler or rewrite this entire portion of code



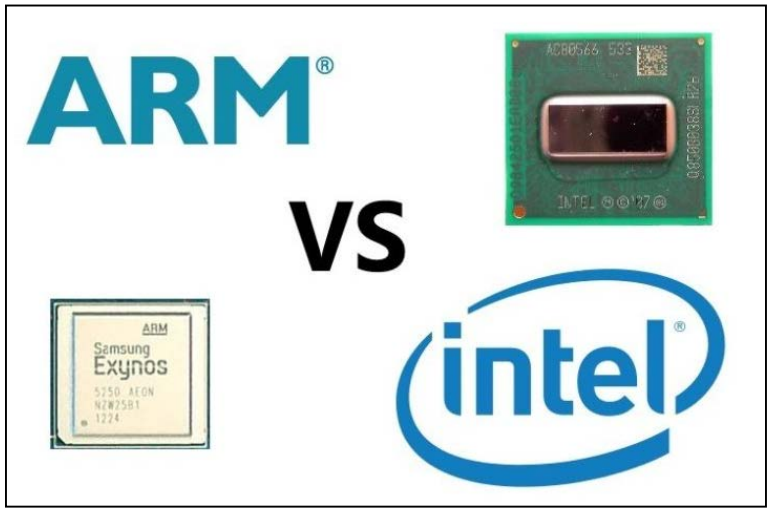
Why is Assembly Language important?
Developing fast and efficient programs

- How can we develop programs that execute as fast as possible?
  - Step 1: developer writes the entire program in a high level language (Examples: C or python)
  - Step 2: identify portions of the code that are ‘timing critical’ that need to be execute as fast as possible
  - Step 3: have the compiler generate assembly code for the ‘timing critical’ portions of code
  - Step 4: the developer can either fine-tune the non-efficient assembly code generated by the compiler or rewrite this entire portion of code

Developers write assembly code so important ‘timing critical’ portions of the program can execute as fast as possible

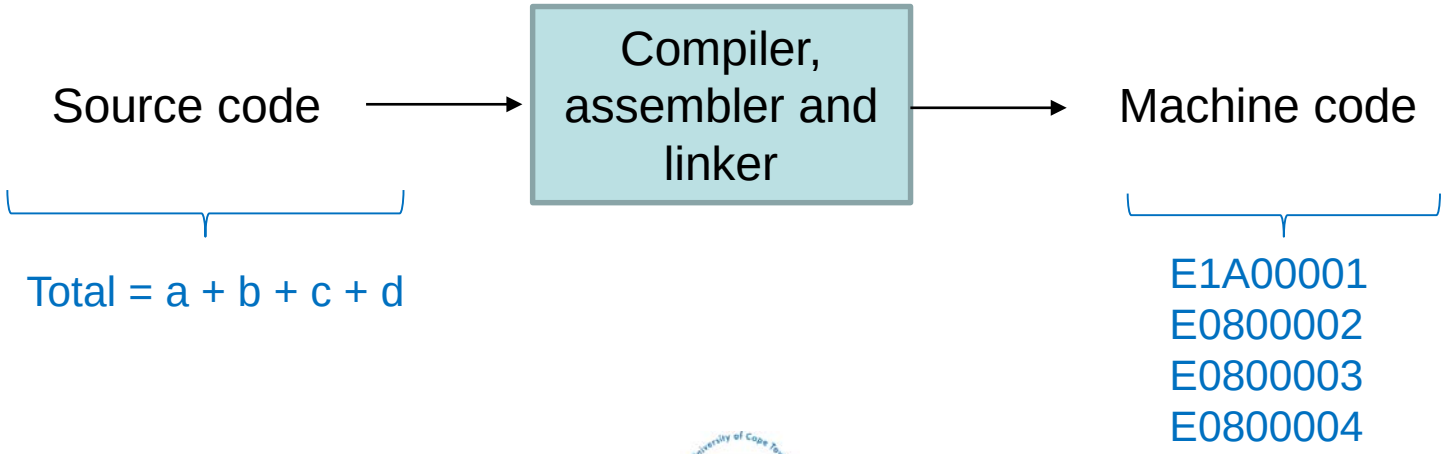
Why is Assembly Language important?
Disadvantages: coding in Assembly language

- Initially, it is long and tedious task to write assembly code
- Assembly code can be fairly difficult to understand and modify
- Code written for one processor may not used for another processor with a different architecture and instruction set.
  Example: ARM CPU versus Intel CPU



Why is Assembly Language important?
Benefits of coding in Assembly language

- Produce fast code to compensate for non-efficient compiler generated code
- Gain a good understanding of how the ALU, CPU registers and memory work together to execute instructions of a program
- Gain skills that can be used to develop compilers, assemblers, linkers, debuggers and other development tools



Why is Assembly Language important?
Approach used to learning Assembly language

1. Understand fundamentals of modern computer architecture
2. Understand basic assembly language instructions
3. Go through the following topics introduced by Ferrer and Pervin:
   a. Ch1: Raspberry Pi Assembler
   b. Ch2: ARM registers
   c. Ch3: Memory
   d. Ch4: Debugging
   e. Ch5: Branching
   f. Ch6: Control structures
   g. Ch7: Addressing modes
   h. Ch8: Arrays and structures
   i. Ch9: Functions
   j. Ch11: Recursion and the Stack