

This activity enables learners to understand how data flows within the central processing unit (CPU) as instructions are executed

Fetch-execute cycle

Introduction

This activity encourages learners to simulate the operation of the fetch-execute cycle. It uses a board and card based activity to help learners understand the basic principles of the process.

Aim

To enable learners to understand how data flows within the central processing unit (CPU) as instructions are executed.

Learning objectives

At the end of this session learners will be able to:

- simulate the actions of the CPU in fetching and executing instructions
- describe the role the various CPU registers play.

Resources

- A board and cards which need to be printed using a colour printer, laminated and cut up to create the playing board and the set of instruction and data cards plus a set of values for the program counter.
- Photocopiable resources are in this booklet. Files can also be downloaded from the Teacher resources CD-ROM under Learner activities, Resources for advanced learning examples.
- Each group will need a board and a set of cards.
- Teacher and trainer guidance.
- Four learner guidance sheets, a different one for each of the roles (PC, ALU, instructions and data).
- PowerPoint presentation stored on the Teacher resources CD-ROM.

Teacher and trainer guidance

Stage 1 - preparation

1. Organise the learners into groups of four.
2. The playing board consists of four pages that are set up in a square on the table. You will need as many copies as there are groups in the class.
3. The Learner guidance sheets contain information about the program file, PC file and data file. Each learner will need a copy of the guidance.
4. Teacher and trainers are strongly advised to work through the activity to prepare for the session.

Stage 2 - Introducing the activity

1. Introduce learners to the ideas behind the fetch-execute cycle. Explain the various registers involved in this process. Explain the task and that this is a very much simplified simulation of what is involved. Explain that they are going to execute a program. Some or all of the slides of the PowerPoint presentation can be used for this part of the session.
2. Break the learners into groups of four and assign the following roles:
 - program counter
 - arithmetic-logic unit
 - instructions
 - data

If they do not divide exactly into groups of four, create some groups of three and assign the roles of instructions and data to one person.
3. Give the person with the program counter role the yellow cards with the numbers 100, 101, 102 and 103.
4. Give the person with the arithmetic-logic role the red card with the number 75.

Hardware and software - Fetch-execute cycle

5. Give the person with the instructions role two copies of blue cards with GET 400, ADD 401, STORE 402, STOP.
6. Give the person with the data role two copies of red cards with the numbers 46 and 29.
7. Set out the playing boards.
8. Give each person the learner guidance sheet that is appropriate to their role and ask them to execute the program.
9. During the activity make sure that the activity is progressing smoothly and be ready to advise if necessary.
10. A summary of the program is included on the next page.

Plenary

At the end of the session summarise the task. Ask learners questions about their experiences and what they found difficult.

A follow up activity might involve them explaining the role of the memory buffer register and/or drawing a diagram showing the data flows between the various registers.

Hardware and software - Fetch-execute cycle

The program

A summary of the program learners are going to execute are given below.

Address	Contents
100	GET 400
101	ADD 401
102	STORE 402
103	STOP
400	46
401	29
402	

Initial state PC=100

Sequence of events

PC	IR	Acc	400	401	402
100	GET 400	46	46	29	0
101	ADD 401	75	46	29	0
102	STORE 402	75	46	29	75
103	STOP	75	46	29	75

Learner guidance for the fetch-execute cycle

Arithmetic-logic unit player (ALU)

The task of your group is to execute a program based on the way a CPU carries out the instruction cycle. The program you are going to execute is given below. Use the flow chart on the table showing the fetch-execute cycle to guide your actions. When it comes to that part of the cycle which is labelled **decode the instruction**, all group members should discuss what this involves for the instruction currently in the instruction register. When you are all agreed, you should carry out the instruction.

Your role is to execute the instruction in the instruction register, which may involve asking for data to be moved between memory and CPU registers. When you need to change the value in a register or a memory location, just put the new value on top of any cards in there from earlier in the program. When you have finished tell the person playing the role of the program counter to start the next cycle. You will be given one data (red) card which you will need to carry out the role of ALU.

The activity comes to an end when you reach the STOP instruction.

Program Add the contents of location 400 to the contents of location 401 and store in location 402

Address	Contents
100	GET 400
101	ADD 401
102	STORE 402
103	STOP
400	46
401	29
402	

Initial state PC=100

Learner guidance for the fetch-execute cycle

Data player

The task of your group is to execute a program based on the way a CPU carries out the instruction cycle. The program you are going to execute is given below. Use the flow chart on the table showing the fetch-execute cycle to guide your actions. When it comes to that part of the cycle which is labelled **decode the instruction**, all group members should discuss what this involves for the instruction currently in the instruction register. When you are all agreed, the person playing the role of the arithmetic-logic unit should carry out the instruction.

You will be given a number of copies of the red data cards which are going to be manipulated by the program. Your first task is to put the data into the right locations in memory; place as many copies of each data value as you have in the right location, one on top of the other. As the activity progresses you will need to respond to requests from the arithmetic-logic unit to give a copy of an item of data (don't forget to ensure that value stays in the location unless it is overwritten) or to overwrite data in your part of memory.

The activity comes to an end when you reach the STOP instruction.

Program Add the contents of location 400 to the contents of location 401 and store in location 402

Address	Contents
100	GET 400
101	ADD 401
102	STORE 402
103	STOP
400	46
401	29
402	

Initial state PC=100

Learner guidance for the fetch-execute cycle

Instructions player

The task of your group is to execute a program based on the way a CPU carries out the instruction cycle. The program you are going to execute is given below. Use the flow chart on the table showing the fetch-execute cycle to guide your actions. When it comes to that part of the cycle which is labelled **decode the instruction**, all group members should discuss what this involves for the instruction currently in the instruction register. When you are all agreed, the person playing the role of the arithmetic-logic unit should carry out the instruction.

You will be given two copies of a number of blue cards representing each instruction in the program. Your first task is to put the instructions of the program into the right locations in memory; place two copies of each instruction in the right memory location, one on top of the other. As the activity progresses you will need to respond to requests from the person playing the role of program counter to give a copy of an instruction in your part of memory.

The activity comes to an end when you reach the STOP instruction.

Program Add the contents of location 400 to the contents of location 401 and store in location 402

Address	Contents
100	GET 400
101	ADD 401
102	STORE 402
103	STOP
400	46
401	29
402	

Initial state PC=100

Learner guidance for the fetch-execute cycle

Program counter player

The task of your group is to execute a program based on the way a CPU carries out the instruction cycle. The program you are going to execute is given below. Use the flow chart on the table showing the fetch-execute cycle to guide your actions. When it comes to that part of the cycle which is labelled **decode the instruction**, all group members should discuss what this involves for the instruction currently in the instruction register. When you are all agreed, the person playing the role of the arithmetic-logic unit should carry out the instruction.

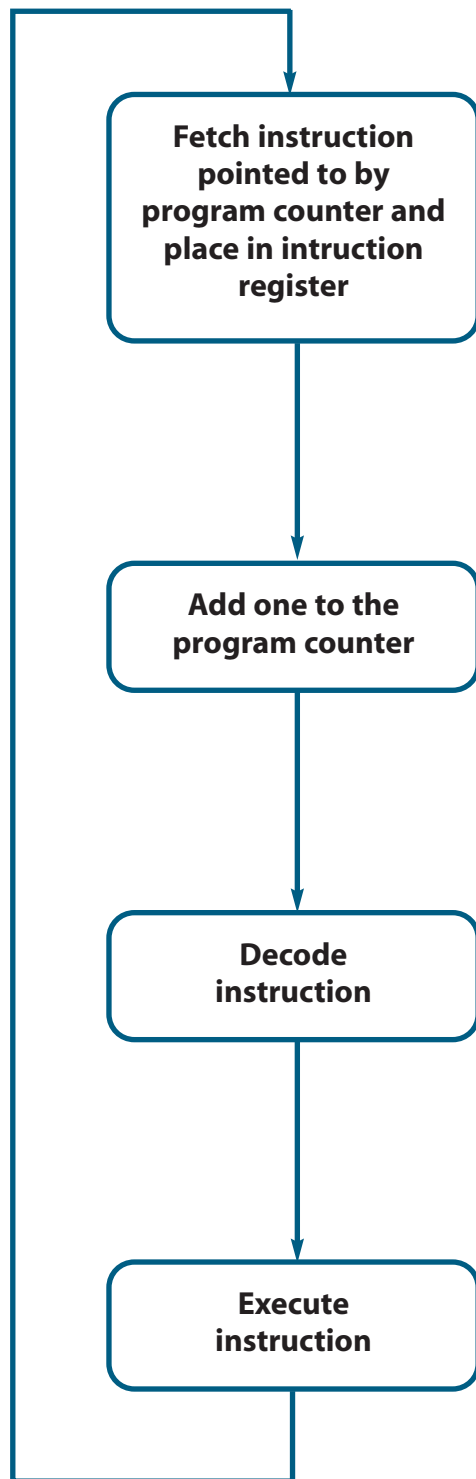
You will be given a number of yellow cards which represent the value of the program counter. Sort them into order and start by placing 100 in the PC register on the playing board and wait for the others to be ready to play. Your task is to start an instruction cycle by asking the person playing the role of instructions to give you the instruction located in the memory address pointed to by the PC. When you get the instruction, place it in the instruction register and update the value in the program counter. Put any new program counter values on top of the ones already in there.

The activity comes to an end when you reach the STOP instruction.

Program Add the contents of location 400 to the contents of location 401 and store in location 402.

Address	Contents
100	GET 400
101	ADD 401
102	STORE 402
103	STOP
400	46
401	29
402	

Initial state PC=100



Instruction set

GET n
copy contents
of location n
to accumulator
register

STORE n
copy contents
of accumulator
register to
location n

ADD n
add the contents
of location n to
the accumulator
register

STOP
stop the program