The **Microelectronics Training Center**

The MTC is an initiative within the INVOMEC division
Industrialization & Training in Microelectronics

Lab-exercise

Lab 4:

Design of the condition check function inside the control unit

Cluster: Cluster1
Module: Module4b

Target group: Students

Version: 1.1
Date: 20/12/06
Author: Osman Allam
Modified by: Geert Vanwijnsberghe
History :

This material was developed with support of the European Social Fund.
ESF: Prevent and combat unemployment by promoting employability, entrepreneurship, adaptability and equal opportunities between women and men, and by investment in people.
http://www.esf-agentschap.be

Introduction

The Control unit is responsible for translating the instructions into a sequence of input signals for all the subunits for a number of clock cycles depending on the instruction. Also the unit that drives the Databus is selected by the Control unit. This Databus is implemented as a multiplexer structure but not shown in the next figure to keep the figure simple. The Databus can be driven by ACC, Stack, MBR and control unit.

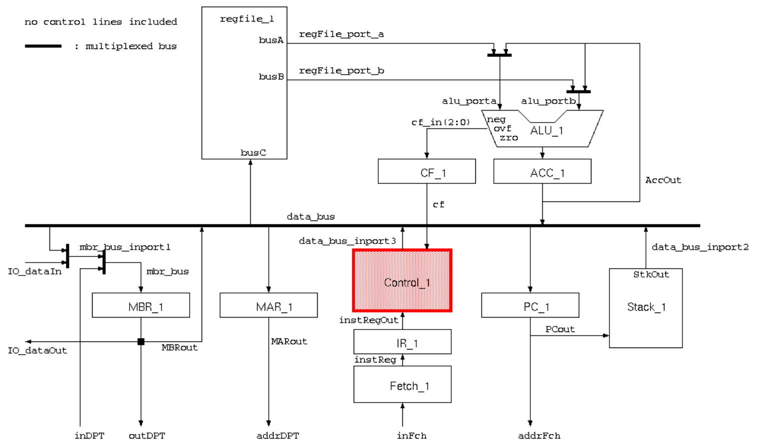

Figure 1: CPU architecture

Objectives

After completing this module, you should be able to:

- Translate logic circuits into VHDL descriptions
- Build systems of different logical units (functions, processes, FSM, etc)
- Understand more complicated testbenches

Knowledge background

- Basic VHDL knowledge
- Understanding of the operation of control units within any computer system

Classification

Level: 3
Duration: 90 minutes

Input

VDHL template

The lab

As mentioned before, the control unit of Micro6 is comprised of 3 sub-units/functions.

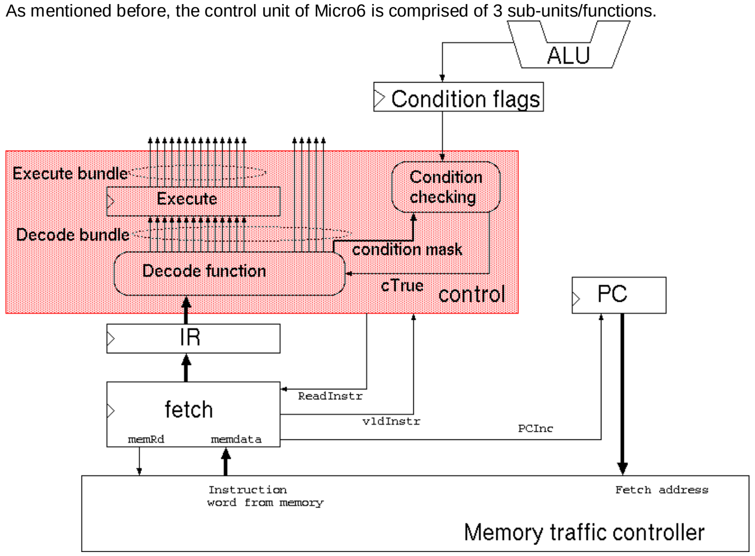

Figure 2

The subunits and their implementation in VHDL are shown in the table below:

Sub-unit	*Implementation*	*VHDL implementation*
Decode unit	Decoder	Function in a package
Execute unit	Moore FSM	3-process FSM
Condition checking unit	Combinational logic	Concurrent signal assignments

The fetch unit will be explained in a later module and is modeled as a separate entity. In this module, we will show how the other sub-units are combined together to build what can be called the decode-execute unit. The separation between the fetch unit and the decode-execute unit emphasizes the structure of the instruction pipeline of Micro6.

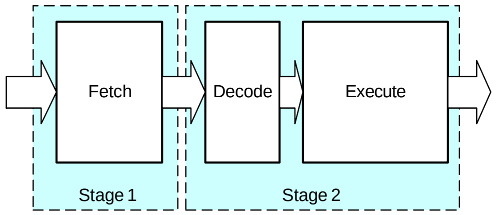

Figure 3: Fetch and decode-execute unit

We would like to emphasize that the control unit will not be implemented as a structural model. However, you still need to declare internal signals to pass information and data between different units. The signals you will definitely need include but are not limited to the following:

1. Decode bundle (including the condition mask);
2. Execute bundle;
3. Condition-true (`cTrue`) signal.

(More info on decode and execute bundle: module 4a)

Condition checking unit/function

This is an auxiliary unit of the control unit. It receives the condition flags from the ALU and matches them with the condition mask provided by branch and jump instructions. The condition checking unit returns an active signal ("1") when the match is successful and an inactive one ("0") otherwise.

Micro6 ALU maintains 3 condition flags:

Bit	*Name*	*description*
2	`neg`	Negative
1	`ovf`	Overflow
0	`zro`	Zero

The condition mask provides information as of which of these flags are to be checked and against what levels (high or low)

Bit	*Name*	*Description*
5	`pn`	Polarity of negative flag
4	`pv`	Polarity of overflow flag
3	`pz`	Polarity of zero flag
2	`cn`	Check negative flag
1	`cv`	Check overflow flag
0	`cz`	Check zero flag

For example, if the `cz` bit is high, the condition checking circuit is required to match the zero flag with the `pz` polarity (level). On the contrary, if the `pv` bit is low, the circuit is NOT required to check the overflow condition.
`cTrue` is active ("1") if and only if all checked condition flags match their corresponding polarities.

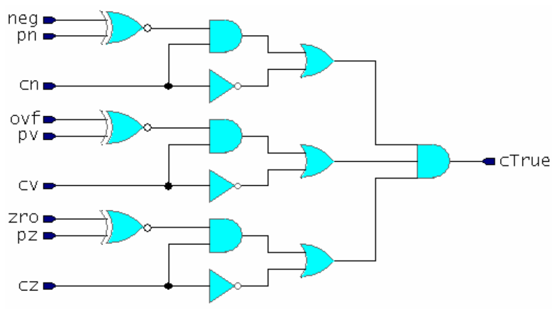

Figure 4: Condition checking circuit

Implement the condition checking unit as described by the circuit above using concurrent signal assignment statements. You may need to declare additional signals for internal nets.

Decode unit/function

The decode function has already been implemented in the previous module as a function in the `micro_control_pk` package.
Incorporate the decode function into the control unit by applying the function `decodeInstr` on the decode unit inputs as shown in the diagram below. Recall that the output of the decode unit is the decode bundle.

Execute unit

The execute unit is the final phase of the control unit generating all the signals to control the data path. It is implemented as a Moore state machine adopting a 3-process style as depicted below.

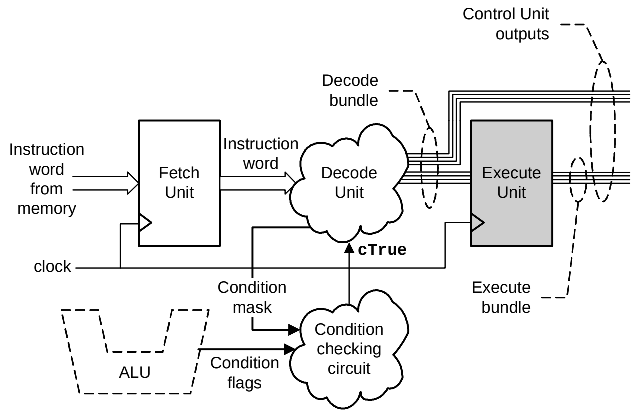

Figure 5

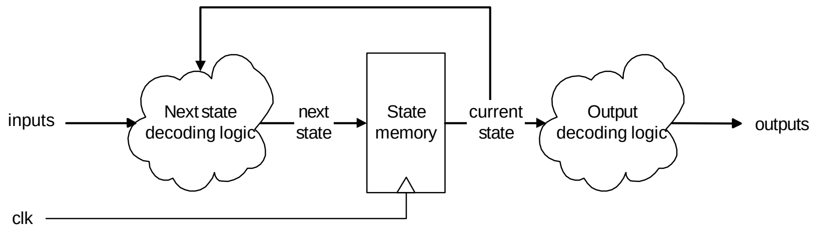

Figure 6: Moore machine

Note that the state machine outputs are not registered. This saves chip area but may decrease the maximum clock frequency of the microprocessor if combinational paths of the control signals are the critical paths of the microprocessor.

Verify your code

Use the tb_control_cc.vhd to verify your implementation of the condition check circuit. The rest of the control unit can partially be verified by the testbench in tb_control.vhd. This code gives you an idea of how such a large controller could be verified at the behavioral level.