CS 250 Fall 2017 Homework 06

Due 11:58pm Thursday, Oct. 19, 2016

Submit your typewritten file in PDF format to Blackboard

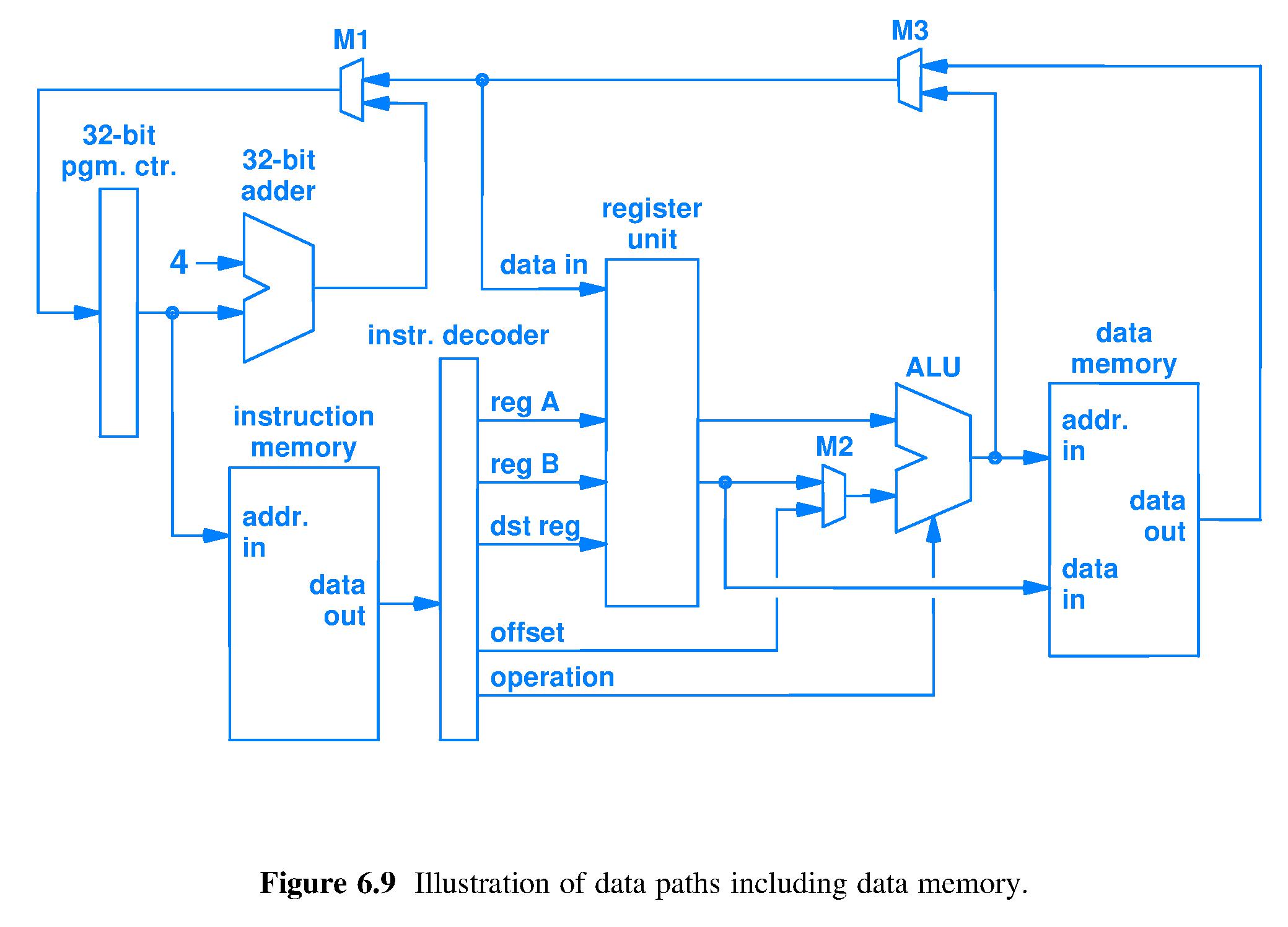
1. Text exercise 6.13. Your answer will have three parts as follows.
   1. Define the new BRR, Branch Relative, instruction by writing a line of C code. Use only labels shown in Figure 6.9 as variable names in your C statement.

ProgramCounter = ProgramCounter + offset

* 1. Design the bit string for BRR and fill in the instruction format table here showing actual bit values where possible, “unused” where appropriate, and your chosen integer representation to be used in the Offset field.  
     Answer:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Opcode | Reg A | Reg B | Dst Reg | Offset |
| bits | 31 – 27 | 26 – 23 | 22 – 19 | 18 – 15 | 14 – 0 |
| BRR | Branch opcode | Operand1 to compare | Operand2 to compare | unused | Signed int for else PC |

* 1. On Figure 6.9, provided below, overlay all the additional circuitry needed to execute BRR, according to your design (there is more than one way to add BRR to the hardware of Fig. 6.9). Use a color other than blue for your circuitry, so that your work clearly stands out.

The new mux is added after the program counter. It chooses the output of the program counter and M1 picks the offset  
  


1. The BRR instruction (see Question 1 above) appearing in assembly as  
     
    .LL2: BRR -36 ; branch to an address -36 (base 10) from the current address  
     
   is part of a machine language instruction loaded into instruction memory. Assume .LL2 corresponds to 0x00400514; the opcode is 5; the computer has byte-addressed, Little Endian memory; and that the assembler used to generate the machine language fills unused fields with zeros.
   1. Fill in the table to show the contents of memory starting at .LL2.

|  |  |
| --- | --- |
| Memory address, 0x representation | Bit string stored, shown in binary representation |
| 0x00400514 | 1010 0000 |
| 0x00400515 | 0000 0000 |
| 0x00400516 | 0000 0000 |
| 0x00400517 | 0000 1101 |

* 1. The computer performs a fetch at address 0x00400514. What bit string is produced at the output wires of the instruction memory? Put a space in your bit string after every fourth bit to help with readability.

1010 0000 0000 0000 0000 0000 0000 1101

* 1. If the instruction memory has 32 wires for output, and the bytes appear on those wires in Little Endian order, how is it that the instruction decoder can receive the machine language bit string fields in their proper arrangement as defined by the instruction format?

If the decoder deals with little endian then nothing happens else the wires are added in reverse.

1. The assembly language for a computer includes the JSR, Jump to SubRoutine, instruction defined as Current\_instruction\_pointer 🡨 Current\_instruction\_pointer + Offset. A subroutine name in a high-level language program is represented in the assembly language code for that program as a label on the first assembly instruction of the subroutine. What does the subroutine name become after the assembler has produced machine language?

Bit Strings

1. Text exercise 9.2.

Near branches and subroutine labels

1. Text exercise 9.9. Also, state the size of the instructions in Figure 9.12.

1 branch for each branch because else will have a branch and it will have a default next of size 32.