*Computer Organization and Architecture*

Marcel Morin   
2606

Uday Sahni  
2606

Richie Piyasirisilp  
2606

Stone Yun  
2606

*Abstract*—This electronic document is a “live” template and already defines the components of your paper [title, text, heads, etc.] in its style sheet. *\*CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract*. (*Abstract*)

Keywords—component, formatting, style, styling, insert (key words)

# Overview

The design problem was to implement a standard five-stage pipelined 32-bit MIPS processor. The proposed processor is able to implement a subset of 27 instructions of the MIPS instruction set architecture. In addition, the processor implements early branching meaning that branch instructions are resolved at the decoding stage rather than the execution stage. This optimization reduces delays imposed by each branch instruction and thereby reduces the performance loss. We will provide a brief overview of each stage in this section.

The instruction fetch stage mainly consists of the program counter, instruction memory, and an ALU. At the beginning of the program, machine code instructions are loaded into the instruction memory module. The ALU and program counter work in tandem to reference the next instruction address. Since each instruction is 32-bits and MIPS uses byte addressing, the ALU adds four to the program counter value to fetch the next instruction. There is also a 2:1 mux that selects between the ALU output and the decoded branch address. This allows our processor to move to another instruction address if our program must branch or jump.

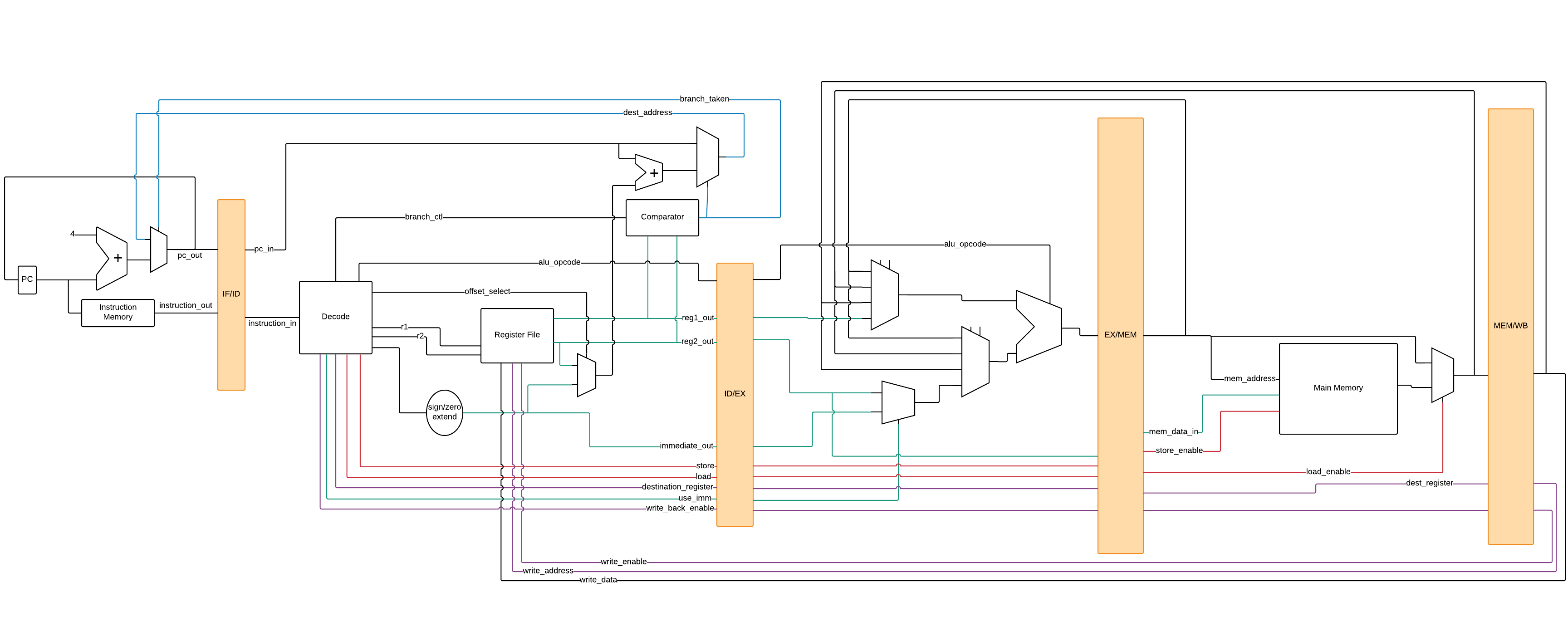
The decode stage mainly consists of our decoder and register file. The decoder interprets the fetched instruction and then outputs the register addresses that the register file will use for referencing and outputting the required data. Since we implemented early branching, branching instructions are resolved in this stage. The decoder outputs the target address of branch instructions and sends it to the comparator, which decides if the program must branch/jump. If the program indeed needs to branch, the target address is added to the current instruction address and sent back to the IF stage for updating the program counter. Finally, there is also a sign/zero extender for sign extending immediate values from 16 bits to 32.

The execute stage mainly consists of a comparator, 4:1 mux’s, a 2:1 mux, and the ALU. The ALU is the most important component of this stage since it performs the arithmetic necessary to execute the instruction. If we are performing a load or store, the ALU adds the offset (forwarded from ID stage) with the register data to create the necessary memory address. If we are performing an add or subtract operation, for example the ALU performs the necessary arithmetic on the data that has been presented to it by the ID stage. The 2:1 mux is used to toggle between feeding the ALU register contents, or an immediate value based on the decoded instruction. It is controlled by the decoder, which outputs a signal to indicate whether the mux should select the register data or the immediate value. Finally, the comparator and 4:1 mux’s implement our data forwarding mechanism. The 4:1 mux’s are connected to the EX, MEM, and WB stage. The comparator checks the register addresses to see if any of the following instruction make references to the same register address that is in the EX, MEM, or WB stages and then sends a control signal to the 4:1 mux’s so that they can select the appropriate register data.

The MEM stage houses our main memory block and either retrieves data on a load instruction or re-writes the referenced memory address on store instructions (in which case, a write enable request has been submitted by the decoder and sent to the memory block). Furthermore, there is a 2:1 mux at the output controlled by the load signal that controls the output of the MEM stage. If we are executing a load instruction, the data from the desired memory location is selected and sent to the register file for write-back. Otherwise, the output of the ALU is selected for writing back to the register file since that data will be the result of executing the instruction.

At the very end of our processor is the WB stage which forwards the resulting data of an instruction to the required location such as the register file and the EX stage (for data forwarding). *Figure 1* shows the high-level organization of our processor. It details the connections and signals within and between each stage.

1. Block diagram of the pipelined processor. Components between the latches (in orange) were organized into module blocks. The top level module manages signals and data that are used by multiple stages. Referenced from an online source [1].



# Procedure

## Design Approach

As this is a large project with five stages and several components per stage, we took a multi-step approach to design. First, we needed a complete block diagram detailing all the components needed for each stage, as seen in *Figure 1*. During the creation of this block diagram, course material and online resources were reviewed. This was to get an idea of which components would be needed and the type of control logic required for implementing features such as data forwarding, branching, and memory addressing (for load and store). After verifying with the course material and online sources that we had all the necessary components and control signals, we wrote the descriptions of these components stage by stage.

The descriptions of all components seen in the block diagram were written from scratch and implemented using VHDL. As we were a group of four, we could do the component design in parallel. Each component is its own module. Upon completion of the components of a stage, they were integrated together as a single block in a higher-level module. This module contains all the input and output signals of its respective stage and connects the ports of each component to the necessary signals. We should note that the latches dividing each of the five stages were written as modules of their own. They acted sort of like “gates” that acted as data control between stages, storing data values from one stage and presenting them to the next stage on the following clock cycle. These latches are essential for the implementation of a pipeline. Storing the results of each stage in a latch frees up that stage to perform subsequent instructions.

Finally, after successfully integrating the components within each stage, a top-level module was designed for connecting the entire pipeline together. In this top-level module, inter-stage signals such as forwarding signals, branching signals, write enabling, and write back data and signals were implemented and connected to their required ports.

## Testing and Evaluation

Identify applicable funding agency here. If none, delete this text box.

The pipelined processor was tested in several stages in ModelSim. The first testing stage was to evaluate each individual component separately to ensure that they behaved as expected. For example, the adder component was tested by forcing its two inputs and its enable signal and verifying that the correct value was outputted. It was important to test each component individually before testing them together because problems could be easily isolated and fixed.

The next testing stage was to evaluate each of the five higher-level pipeline stage modules with a test bench. This was done to verify that given the correct inputs and signals, each pipeline stage would be able to produce the correct outputs. For example, the test bench written for the instruction fetch stage set the branch\_taken variable to force which instruction would be performed next. The program counter and destination address inputs were tracked to check that they only changed when they were expected to, depending on the test case. The pc\_out and instruction\_out outputs were also tracked to ensure that the fetching stage produced the appropriate program count and instruction to be fed into the decoding stage.

Lastly, all five stages were tested together to evaluate the pipelined processor as a whole. Similarly to the individual pipeline stages, the final product was tested using a test bench. A set of MIPs instructions composed of arithmetic, logical, transfer, shift, memory, and control-flow operations was inputted and processed in the pipeline processor. Variable delay cycles were also added between the instructions to provide a more detailed test bench. The results were then written to a .txt file that was used to verify that all the operations and procedures performed as expected.

Even though the entire pipelined processor was tested from each individual component to full stages, there were still multiple problems when testing all five stages together. The two main issues that persisted were in the port mapping and latency between stages. To fix the issues in the port mapping, every associated component was checked and remapped so that all the wiring connected correctly. As for the latency problems, the components in question were reviewed and were found to have design problems that lead to timing errors.

# Optimizations

Before you begin to format your paper, first write and save the content as a separate text file. Complete all content and organizational editing before formatting. Please note sections A-D below for more information on proofreading, spelling and grammar.

Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

## Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

## Units

* Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as “3.5-inch disk drive”.
* Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.
* Do not mix complete spellings and abbreviations of units: “Wb/m2” or “webers per square meter”, not “webers/m2”. Spell out units when they appear in text: “. . . a few henries”, not “. . . a few H”.
* Use a zero before decimal points: “0.25”, not “.25”. Use “cm3”, not “cc”. (*bullet list*)

## Equations

The equations are an exception to the prescribed specifications of this template. You will need to determine whether or not your equation should be typed using either the Times New Roman or the Symbol font (please no other font). To create multileveled equations, it may be necessary to treat the equation as a graphic and insert it into the text after your paper is styled.

Number equations consecutively. Equation numbers, within parentheses, are to position flush right, as in (1), using a right tab stop. To make your equations more compact, you may use the solidus ( / ), the exp function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash rather than a hyphen for a minus sign. Punctuate equations with commas or periods when they are part of a sentence, as in:

*a**b* 

Note that the equation is centered using a center tab stop. Be sure that the symbols in your equation have been defined before or immediately following the equation. Use “(1)”, not “Eq. (1)” or “equation (1)”, except at the beginning of a sentence: “Equation (1) is . . .”

## Some Common Mistakes

* The word “data” is plural, not singular.
* The subscript for the permeability of vacuum **0, and other common scientific constants, is zero with subscript formatting, not a lowercase letter “o”.
* In American English, commas, semicolons, periods, question and exclamation marks are located within quotation marks only when a complete thought or name is cited, such as a title or full quotation. When quotation marks are used, instead of a bold or italic typeface, to highlight a word or phrase, punctuation should appear outside of the quotation marks. A parenthetical phrase or statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.)
* A graph within a graph is an “inset”, not an “insert”. The word alternatively is preferred to the word “alternately” (unless you really mean something that alternates).
* Do not use the word “essentially” to mean “approximately” or “effectively”.
* In your paper title, if the words “that uses” can accurately replace the word “using”, capitalize the “u”; if not, keep using lower-cased.
* Be aware of the different meanings of the homophones “affect” and “effect”, “complement” and “compliment”, “discreet” and “discrete”, “principal” and “principle”.
* Do not confuse “imply” and “infer”.
* The prefix “non” is not a word; it should be joined to the word it modifies, usually without a hyphen.
* There is no period after the “et” in the Latin abbreviation “et al.”.
* The abbreviation “i.e.” means “that is”, and the abbreviation “e.g.” means “for example”.

An excellent style manual for science writers is [7].

# Conclusions

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

## Authors and Affiliations

**The template is designed for, but not limited to, six authors.** A minimum of one author is required for all conference articles. Author names should be listed starting from left to right and then moving down to the next line. This is the author sequence that will be used in future citations and by indexing services. Names should not be listed in columns nor group by affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization).

### For papers with more than six authors: Add author names horizontally, moving to a third row if needed for more than 8 authors.

### For papers with less than six authors: To change the default, adjust the template as follows.

#### Selection: Highlight all author and affiliation lines.

#### Change number of columns: Select the Columns icon from the MS Word Standard toolbar and then select the correct number of columns from the selection palette.

#### Deletion: Delete the author and affiliation lines for the extra authors.

## Identify the Headings

Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

Component heads identify the different components of your paper and are not topically subordinate to each other. Examples include Acknowledgments and References and, for these, the correct style to use is “Heading 5”. Use “figure caption” for your Figure captions, and “table head” for your table title. Run-in heads, such as “Abstract”, will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head from the text.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced. Styles named “Heading 1”, “Heading 2”, “Heading 3”, and “Heading 4” are prescribed.

## Figures and Tables

#### Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation “Fig. 1”, even at the beginning of a sentence.

1. Table Type Styles

| Table Head | Table Column Head | | |
| --- | --- | --- | --- |
| Table column subhead | Subhead | Subhead |
| copy | More table copya |  |  |

1. Sample of a Table footnote. (*Table footnote*)
2. Example of a figure caption. (*figure caption*)

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity “Magnetization”, or “Magnetization, M”, not just “M”. If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write “Magnetization (A/m)” or “Magnetization {A[m(1)]}”, not just “A/m”. Do not label axes with a ratio of quantities and units. For example, write “Temperature (K)”, not “Temperature/K”.

##### Acknowledgment *(Heading 5)*

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

##### References

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first ...”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

1. G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*

We suggest that you use a text box to insert a graphic (which is ideally a 300 dpi TIFF or EPS file, with all fonts embedded) because, in an MSW document, this method is somewhat more stable than directly inserting a picture.

To have non-visible rules on your frame, use the MSWord “Format” pull-down menu, select Text Box > Colors and Lines to choose No Fill and No Line.

1. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
2. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
3. K. Elissa, “Title of paper if known,” unpublished.
4. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
5. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
6. Figure 1 reference