Hardware Design of

Real-time Classification Algorithm

*TEAM POOR HANDWRITING*

Haining Qiu, Harry Zhao, Justin Qiao, Qikun Liu

A picture containing logo

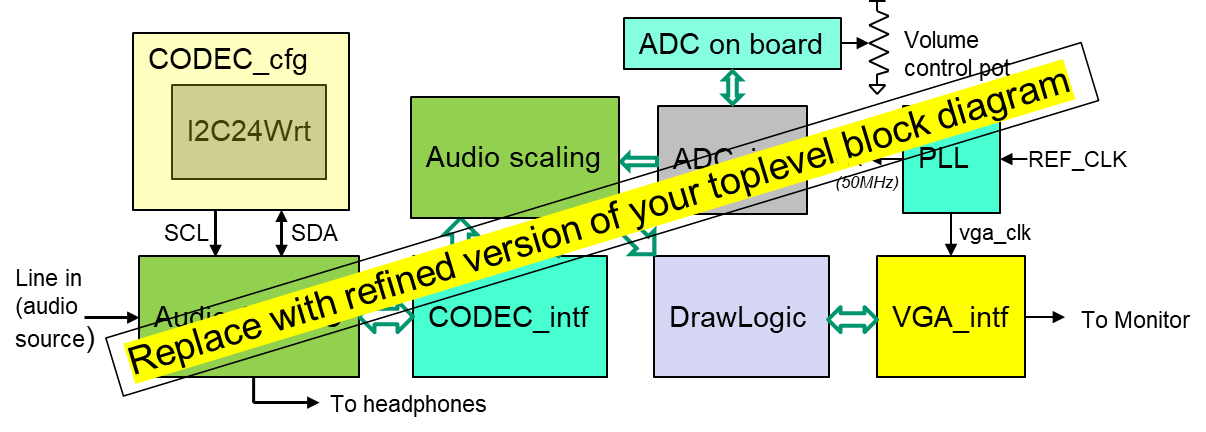
Description automatically generated

Interface Document

1. Hardware Block Diagram
   1. CPU
      1. PC
   2. Extended ALU
      1. Floating-point Adder
      2. Floating-point Multiplier
      3. Float-to-integer Unit
      4. Integer-to-float Unit
      5. 16-by-16 Integer Multiplier
   3. Stack
   4. Image Compressor and Storage
2. Software

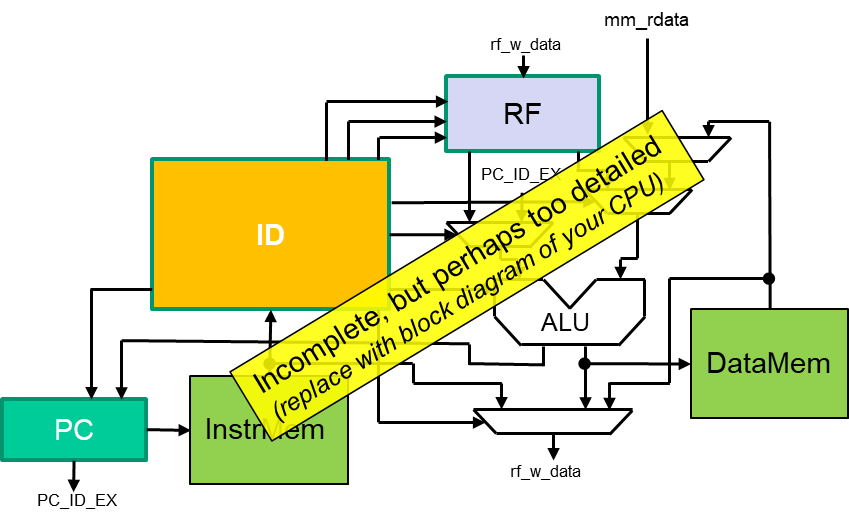
# Hardware Block Diagrams

Refine the hardware block diagram in this document. You may need separate block diagrams for some of the more complex functional units. The CPU would fall into this category. Each unit that you are defining interfaces for should be a “block” in a presented block diagram. CPU block is too complicated to write as an interface. In addition to signal interface we wish to see a description of any significant registers within a unit.



## CPU

One complex functional unit we all have that needs further sub-dividing is the CPU. Show a block diagram of your CPU below:



### PC (Program Counter) Interface

|  |  |  |
| --- | --- | --- |
| **Signal:** | **Dir:** | **Description:** |
| Int\_occurred | In | Indicates an interrupt occurred and should jump to interrupt vector provided **int\_vec. PC** should be stored to context save copy |
| Int\_vec[15:0] | In | From interrupt controller. Specifies ISR location to jump to |
| Flow\_change\_ID\_EX | In | Indicates PC will be updated due to branch or jump (from ID unit) |
| Rti\_ID\_EX | in | Return from interrupt has occurred. Restore context saved PC |
| Stall\_IM\_ID | in | Asserted if stalling pipe due to load/use or MOVC related stall |
| Dst\_ID\_EX[15:0] | In | Branch target address from ALU |
| Pc[15:0] | out | Forms address to instruction memory |
| Pc\_ID\_EX[15:0] | out | Piped version needed in EX stage for computing new branch target |
| PC\_EX\_DM[15:0] | out | Piped to EX\_DM for JAL instruction store in R15 |

### PC (Program Counter) Registers

|  |  |
| --- | --- |
| Register: | Description: |
| Pc,PC\_ID\_EX,PC\_EX\_DM  All 16-bits | Described in table above |
| Pc\_saved[15:0] | Context store of next PC saved/restored during interrupts/RTI |
| Flow\_change\_last\_stalled | A MOVC followed by a branch can cause a scenario where flow change and stall can happen simultaneously. In which case we need to update PC with newPC\_stalled next |
| newPC\_stalled[15:0] | Capture of target address if stalled. |

For each block capture it interface and any significant register in tables as shown above for PC.

## Extended ALU

The extended ALU is in the same pipeline stage of the original ALU in the processor while its hardware supports floating-point operations and integer multiplication. It contains five submodules for floating-point addition and multiplication, conversions between float and integer, and integer multiplication. Its output value and flags are selected by the func signal.

### Floating-point Adder

|  |  |  |
| --- | --- | --- |
| **Signal:** | **Dir:** | **Description:** |
| A[31:0] | in | 32-bit input interpreted as an IEEE-754 floating-point number |
| B[31:0] | in | 32-bit input interpreted as an IEEE-754 floating-point number |
| Out[31:0] | out | 32-bit output as an IEEE-754 floating-point number |

### Floating-point Multiplier

|  |  |  |
| --- | --- | --- |
| **Signal:** | **Dir:** | **Description:** |
| A[31:0] | in | 32-bit input interpreted as an IEEE-754 floating-point number |
| B[31:0] | in | 32-bit input interpreted as an IEEE-754 floating-point number |
| OUT[31:0] | out | 32-bit output as an IEEE-754 floating-point number |

### Float-to-integer Unit

|  |  |  |
| --- | --- | --- |
| **Signal:** | **Dir:** | **Description:** |
| FP\_val[31:0] | in | 32-bit input interpreted as an IEEE-754 floating-point number |
| signed\_int\_val[31:0] | out | 32-bit output converted into a signed integer |

### Integer-to-float Unit

|  |  |  |
| --- | --- | --- |
| **Signal:** | **Dir:** | **Description:** |
| signed\_int\_val[31:0] | in | 32-bit input interpreted as a signed integer |
| FP\_val[31:0] | out | 32-bit output converted into an IEEE-754 floating-point number |

### 16-by-16 Integer Multiplier

|  |  |  |
| --- | --- | --- |
| **Signal:** | **Dir:** | **Description:** |
| A[31:0] | in | 32-bit input interpreted as an integer |
| B[31:0] | in | 32-bit input interpreted as an integer |
| sign | in | 1 for signed multiplication; 0 for unsigned multiplication |
| OUT[31:0] | out | 32-bit output as an integer |

## 1.3. Stack

## 1.4. Image Processing and Storage

### Image Compressor

|  |  |  |
| --- | --- | --- |
| **Signal:** | **Dir:** | **Description:** |
| clk | in | 25MHz clock signal from VGA display |
| rst\_n | in | Reset signal |
| start | in | Signals a valid pixel color input starting from 0 |
| pix\_color\_in[7:0] | in | 8-bit pixel color (0 to 255 grey scale) |
| pix\_haddr[7:0] | in | Asserted if stalling pipe due to load/use or MOVC related stall |
| pix\_vaddr[7:0] | in | Branch target address from ALU |
| sram\_wr | out | Forms address to instruction memory |
| pix\_color\_out | out | Piped version needed in EX stage for computing new branch target |
| compress\_addr | out | Piped to EX\_DM for JAL instruction store in R15 |

# Software

It is difficult for us to generalize what software your project entails. All projects involve the firmware running on the processor. This section should include a description of this software, and how it interfaces with the hardware elements.

Some projects also involve software external to the hardware system (learning software for ML, simulator for processor, compiler for processor, …). Please provide a brief description of this software and how it integrates or affects the entire system.