Simple Monolength Instruction Set (SMIS)

User Manual

Architecture overview

REGISTERS

4-bit addressing, supporting up to 16 general-purpose registers
16-bit data, supporting an unsigned value of up to 65,535
R0 is a constant zero value, can be referenced with RZR
R15 is conventionally used for stack pointer, can be referenced with RSP
R14 is conventionally used for base pointer, can be referenced with RBP
R13 is conventionally used for link register, can be referenced with RLR

MEMORY

16-bit addressing, supporting up to 65,535 halfwords of memory
16-bit data, supporting an unsigned value of up to 65,535

FLAGS

Zero Flag (ZF) is set on any arithmetic instruction if the result is 0
Sign Flag (SF) is set on any arithmetic instruction if the result has a 1 in its MSB

SYNTAX

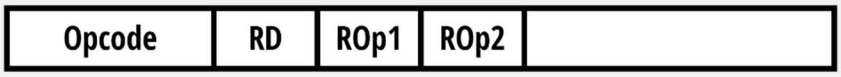
Registers: R<number 0-15>
Immediates: #<16-bit unsigned value>
Comments: //<comment text>
Labels: <label name>:

NOTES

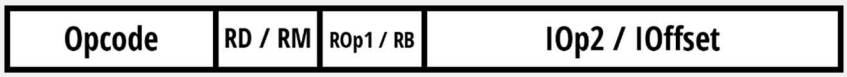
All opcodes are 8 bits
Program is always assumed to begin at memory address 0x0

```
[I] SET                 <dest reg>          <immediate>
[R] COPY                <dest reg>          <source reg>

[R] ADD                 <dest reg>          <op1 reg>           <op2 reg>
[R] SUBTRACT            <dest reg>          <op1 reg>           <op2 reg>
[R] MULTIPLY            <dest reg>          <op1 reg>           <op2 reg>
[R] DIVIDE              <dest reg>          <op1 reg>           <op2 reg>
[R] MODULO              <dest reg>          <op1 reg>           <op2 reg>

[R] COMPARE             <op1 reg>           <op2 reg>

[R] SHIFT-LEFT          <dest reg>          <op1 reg>           <shift reg>
[R] SHIFT-RIGHT         <dest reg>          <op1 reg>           <shift reg>

[R] AND                 <dest reg>          <op1 reg>           <op2 reg>
[R] OR                  <dest reg>          <op1 reg>           <op2 reg>
[R] XOR                 <dest reg>          <op1 reg>           <op2 reg>
[R] NAND                <dest reg>          <op1 reg>           <op2 reg>
[R] NOR                 <dest reg>          <op1 reg>           <op2 reg>
[R] NOT                 <dest reg>          <op1 reg>

[I] ADD-IMM             <dest reg>          <op1 reg>           <op2 imm>
[I] SUBTRACT-IMM        <dest reg>          <op1 reg>           <op2 imm>
[I] MULTIPLY-IMM        <dest reg>          <op1 reg>           <op2 imm>
[I] DIVIDE-IMM          <dest reg>          <op1 reg>           <op2 imm>
[I] MODULO-IMM          <dest reg>          <op1 reg>           <op2 imm>

[I] COMPARE-IMM         <op1 reg>           <op2 imm>

[I] SHIFT-LEFT-IMM      <dest reg>          <op1 reg>           <shift imm>
[I] SHIFT-RIGHT-IMM     <dest reg>          <op1 reg>           <shift imm>

[I] AND-IMM             <dest reg>          <op1 reg>           <op2 imm>
[I] OR-IMM              <dest reg>          <op1 reg>           <op2 imm>
[I] XOR-IMM             <dest reg>          <op1 reg>           <op2 imm>
[I] NAND-IMM            <dest reg>          <op1 reg>           <op2 imm>
[I] NOR-IMM             <dest reg>          <op1 reg>           <op2 imm>

[I] LOAD                <dest reg>          <base reg>          <offset imm>
[I] STORE               <src reg>           <base reg>          <offset imm>

[J] JUMP                <dest label>
[J] JUMP-IF-ZERO        <dest label>
[J] JUMP-IF-NOTZERO     <dest label>
[J] JUMP-LINK           <dest label>

[J] HALT
```

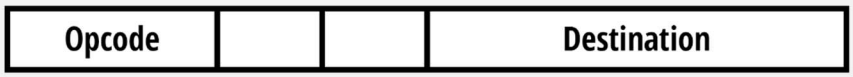
R(egister)-type: 4-bit destination register, 4-bit operand 1 register, 4-bit operand 2 register



I(mmediate)-type: 4-bit arithmetic destination/memory source or destination register,
4-bit operand 1 or base address register, 16-bit operand 2 or offset immediate



J(ump)-type: 16-bit destination immediate assembled from label



Clarifications:

- RD means “Register Destination.”
- ROp1/2 mean “Register Operand 1/2."
- RM means “Memory Register” – can either be source or destination for STORE or LOAD, respectively. These are usually referenced individually as “RSrc” or “RDest.”
- RB means “Register Base Address,” used in STORE and LOAD.
- IOp2 means “Immediate Operand 2.” In SET, this can also just be referred to as “IVal,” for “Immediate Value,” as the instruction does not have a first operand.
- IOffset means “Immediate Address Offset,” used in STORE and LOAD.
- Destination is the destination address of all J-Type instructions.