sml2008-am01: Decoded Instruction Format

Adam Megacz

October 31, 2008

Abstract

This memo gives the decoded, 37-bit-wide, mostly-one-hot format used internally within the dock circuitry to represent an instruction.

Changes:

21-Aug	Initial Revision
23-Oct	Changed polarity of bit 20 on "Shift" and "Set Data Latch"
	Noted that "Immediate→ILC" must have bit 7 set to 0
	Labeled bits 9 and 7 on last two instruction forms
29-Oct	Added TAIL instruction
	Removed "done" bit, relocated infinity bit
30-Oct	Divided `move` instruction into subinstructions based on path latch
31-Oct	Added encoding of `Predicate` field
01-Nov	Changed `Rq` to $\overline{\texttt{OS}}$
	Changed `Int` to $\overline{\texttt{Int}}$
	Swapped Z and !Z

Overview

FleetTwo Instructions in main memory occupy 37 bits. Of this, 11 bits give the path to the dock which is to execute the instruction; thus, only 26 of these bits are interpreted by the dock.

It is easiest to design the OD and EX stages of the dock if the control bits supplied there are mostly one-hot encoded. Moreover, due to layout considerations there is very little cost associated with making the instruction fifo 36 bits wide rather than 26 bits wide.

Due to these two considerations, all 26-bit instructions binary-coded-control instructions are expanded into 36-bit unary-coded-control instructions upon entry to the instruction fifo. This memo documents the 36-bit unary-coded-control format.

Predicate Field

The `Predicate` field, common to many instructions, consists of a six-bit wide, one-hot encoded field. The instruction will be **skipped** (not executed) if **any** condition corresponding to a bit whose value is one is met.

36	35	34	33	32	31	30		1
Z	!Z	!B	B	!A	A			

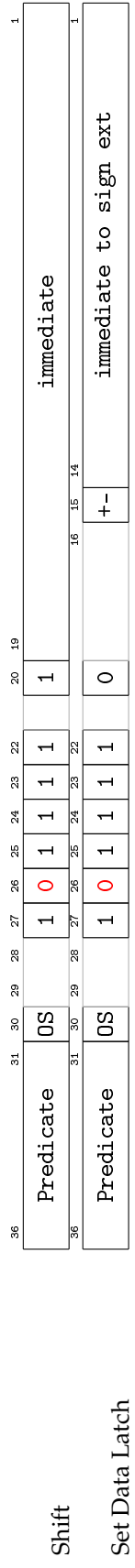The `Z` flag is an "imaginary" flag which is "set" iff the outer loop counter is zero.

For example, if bits 31 and 34 are set, the instruction will be skipped if either the `B` flag is cleared or the `A` flag is set. Equivalently, it will be executed iff the `B` flag is set and the `A` flag is cleared.
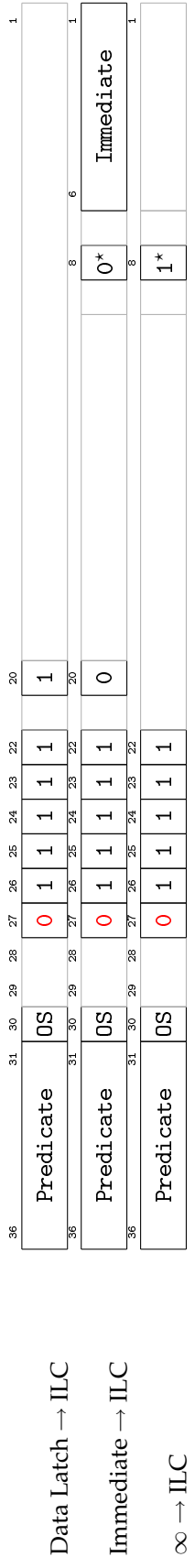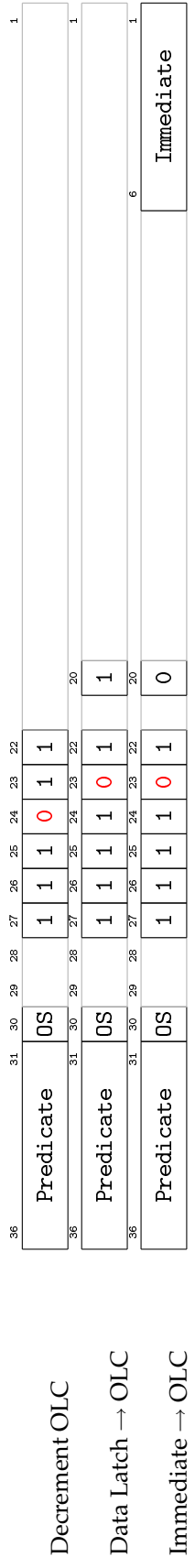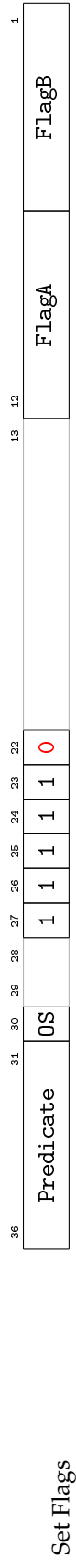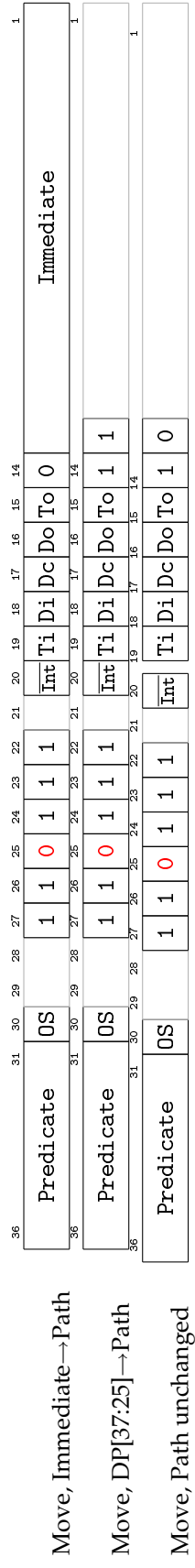
Legend

```
OS = One-Shot (0=Requeueing, 1=Not-Requeueing)
Int = Not Interruptible (0=Torpedoable, 1=Not-Torpedoable)
```

Shift

31 Predicate	30 OS	27	26	25	24	23	22	20	19 immediate	1
Predicate	OS	1	0	1	1	1	1	1	immediate	1

Set Data Latch

31 Predicate	30 OS	27	26	25	24	23	22	20	15/14 +-	immediate to sign ext	1
Predicate	OS	1	0	1	1	1	1	0	+-	immediate to sign ext	1

Move, Immediate→Path

31 Predicate	30 OS	27	26	25	24	23	22	20 Int̄	19 Ti	18 Di	17 Dc	16 Do	15 To	14	13..12 Immediate	1
Predicate	OS	1	1	0	1	1	1	Int̄	Ti	Di	Dc	Do	To	0	Immediate	1

Move, DP[37:25]→Path

31 Predicate	30 OS	27	26	25	24	23	22	20 Int̄	19 Ti	18 Di	17 Dc	16 Do	15 To	14	1
Predicate	OS	1	1	0	1	1	1	Int̄	Ti	Di	Dc	Do	To	1	1

Move, Path unchanged

31 Predicate	30 OS	27	26	25	24	23	22	20 Int̄	19 Ti	18 Di	17 Dc	16 Do	15 To	14	0
Predicate	OS	1	1	0	1	1	1	Int̄	Ti	Di	Dc	Do	To	1	0

Set Flags

31 Predicate	30 OS	27	26	25	24	23	22	13..12 FlagA	FlagB	1
Predicate	OS	1	1	1	1	1	0	FlagA	FlagB	1

Decrement OLC

31 Predicate	30 OS	27	26	25	24	23	22	20	1
Predicate	OS	1	1	1	0	1	1	1	1

Data Latch → OLC

31 Predicate	30 OS	27	26	25	24	23	22	20	1
Predicate	OS	1	1	1	1	0	1	1	1

Immediate → OLC

31 Predicate	30 OS	27	26	25	24	23	22	20	6 Immediate	1
Predicate	OS	1	1	1	1	0	1	0	Immediate	1

Data Latch → ILC

31 Predicate	30 OS	27	26	25	24	23	22	20	1
Predicate	OS	0	1	1	1	1	1	1	1

Immediate → ILC

31 Predicate	30 OS	27	26	25	24	23	22	20	8 0⋆	6 Immediate	1
Predicate	OS	0	1	1	1	1	1	0	0⋆	Immediate	1

∞ → ILC

31 Predicate	30 OS	27	26	25	24	23	22	8 1⋆	1
Predicate	OS	0	1	1	1	1	1	1⋆	1

TAIL

21	
1	

⋆ – bit 8 is the "infinity" bit