Continue ISA Generalization

Three different instruction set architectures are being studied. These three -- L/S, R/M, R+M -- differ in the kinds of operand reference allowed in the primitive instructions.

Conventions for Mnemonics

Adopt conventions: **<op>.<data_type>**

ADD	Addition
SUB	Subtraction
MPY	Multiplication
DIV	Division
INC	Increment
DEC	Decrement
CMP	Compare
CLR	Clear
LD	Load
ST	Store
LDA	Load Address
MOVE	Move reg or mem
SHL	Shift left
SHR	Shift right
BR	Branch Unconditionally
BC	Conditinal Branch
BAL	Branch and Link
BALR	Branch to Reg and Link

B	Byte (8 bits)
UB	Unsigned byte
H	Half word (16 bits)
UH	Unsigned half word
W	Word (32 bits)
UW	Unsigned word
F	Floating point (32 bits)
D	Double precision fp (64 bits)
C	Character or Decimal (8 bits)
P	Packed Decimal (4-bit)

Conventions on Branching

BR	TARGET	Unconditional branch to target
BC	TARGET	Conditional branch w/o condition specified
BC.CC	TARGET	Conditional branch w/o condition specified
BC.NE	TARGET	Conditional branch to target with CC given
BCT.NE	R1,TARGET	Decrement R1 & branch to target if result ≠ 0
BAL	TARGET	Save PC in implied reg & unconditional branch
BALR	R1,TARGET	Save PC in implied reg & branch addr in R1

CC	Condition
T	True (always)
F	False (never)
V	Overflow
C	Carry or Borrow
PE	Even parity
PO	Odd parity

CC	Condition
LE	Less than or equal
LT	Less than
EQ	Equal
NE	Not equal
GE	Greater than or equal
GT	Greater than

Conventions on Operands

Mode	Spec	Meaning
Register	Rx	Register x
Memory	ADDR	Address specified by ADDR
Indirect	[Rx]	Address specified by contents of Rx
Indexed	OFFSET[Rx]	Address give by OFFSET + contents of Rx
Immediate	#*val*	The value *val*

ADD.x R_d, R_{s1}, R_{s2}

0[Rx] = [Rx]

X = X[0]

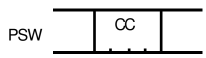
#*val*d	The decimal value *val*
#*val*h	The hexadecimal value *val*
#*val*b	The binary value *val*
#*val*q	The octal value *val*
'*char*'	The value of the character *char*

Program Status Word

The critical state and control information for an executing program is collected together in the program status word -- its usually quite a bit larger than a word.

Typical Contents

- Condition code
- Program Counter (PC)
- Current instruction length
- Enable/disable for arith overflow
- Odd/even parity information
- User ID
- Protection information
- Wait/run state
- Machine check mask enable -- diagnostic
- I/O channel masks



Positive result	-->	set 1000
Negative result	-->	set 0100
Zero result	-->	set 0010
Overflow	-->	set 0001

To test greater than or equal, Mask =1010

Assembly Code

Compute N! using different architectural styles.

```
fact:   MOVE.W  R8,R0       ; Clear upper 1/2 of reg. pair
        LDA     R9, #1      ; Initialize lower 1/2 to 1
        LD.W    R10,N       ; Fetch n
loop:   MPY.W   R8,R9,R10   ; Compute product
        SUB.W   R10,#1      ; Decrement count
        BC.GT   loop        ; Iterate
        ST.W    NFACT       ; Save result
        END
```

```
fact:   ST.W    NFACT,#1    ; Initialize result
        LD.W    R8,N        ; Fetch n
loop    MPY.W   NFACT,R8    ; Compute product
        BCT.NE  loop        ; Iterate
        END
```


Machine Formats

Different operand types require different size encoding.