2/3/65

282

FØRTRAN IV

Standard Arithmetic Package. %FAP

There will be a standard FORTRAN package with the following characteristics which will be compiled into any program which contains a FORTRAN segment.

In the following list of entries. X, I are in ACC

Y, J addresses in X3

or n in X3

New no.	Old no. (struck)	Entry
8	0	I**J ~~Acc~~
9	1	I**n
10	2	X**J
11	3	X**n
12	4	J**I
13	5	n**I
14	6	Y**I
21	7	I/J
22	8	I/n
23	9	J/I
24	10	n/I
6	11	Array
15	12	I → X
16	13	J → %LIB (Y)
17	14	n → %LIB (Y)
18	15	X → I
20	16	Array (from length
19	17	Trace Entry.

Double Precision Package ~~%DPA~~ ~~%FDA~~ %FDP

	0	X+Y		
	1	X*Y		
	2	X−Y		
	3	Y−X	7	Real → DP (Y)
	4	−X		
	5	X/Y		
	6	Y/X		
%FDZ →	0	X**J		
	1	X**n		
	2	Y**I		
	~~10~~	~~Real → DP (Acc)~~		
	~~11~~	~~Real → DP. (n)~~		
	~~12~~	~~DP → Real Acc.~~		

Name in 4 chars in DPAR1 4H%FDP

DPAR2 4H%FDZ

Complex Arithmetic Package %FCA

Entry Points		
	0	X+Y
	1	X*Y
	2	X−Y
	3	Y−X
	4	−X
	5	X/Y
	6	Y/X
	7	Real→Cplx (Y)
%FCZ	0	X**J
	1	X**n
	2	Y**I
	~~10~~	~~Real→Cplx (Acc)~~
	~~11~~	~~Real→Cplx (n)~~

X = Complex No. in Acc

Y = " " address in X3

I = Integer in Acc

J = " address in X3

n = Integer in X3 (<4096)

Name in 4 chars in CPAR1 4H%FCA

CPAR2 4H%FCZ

Exponentiation Routines

Real Exponentiation %FEX4

Entry	0	X ** Y	X in <u>Acc</u>
	1	Y ** X	Y address in X3

Double Precision ** Real %FDR

FDRE 4H%FDR	Entry	0	X ** Y	X in <u>Acc</u>
		1	Y ** X	Y in store. (address in X3)

Double Precision ** D.P %FDD

	Entry	0	X ** Y	X in <u>Acc</u>
FDDE 4H%FDD		1	Y ** X	Y address in X3

NWRD+4 = 0

REL+4 = "AREL"

Standard Binary Operations   64 operations.   0→63

0.   Int-Int.   Acc ~~op~~ n (Direct) +   Acc = X6   ~~Rel~~

Operation	Name	Instr			Rel	Count	Result
Add.	AII ✓	ADX	6	0	Rel = 1	~~2~~ 1	IACC + n
Subt.	SII ✓	SBX	6	0	1	1	IACC − n
mpy	MII ✓	MPY	6	0	1	2	IACC * n
		SLA	67	23	NØRE+1		
		100	3	0	1		
div.	✓ DII	CALL ~~DVS~~	~~5~~ 1	~~8~~ 21	AREL+1 ~~5~~	2	IACC/n
Add (ind)	AIII ✓	ADX	6	0(3)	NØRE+1	1	IACC + n(ind)
Subt (ind)	SIII ✓	SBX	6	0(3)	NØRE+1	1	IACC − n(ind)
mpy (ind)	MIII ✓	MPY	6	0(3)	NØRE+1	2	IACC * n(ind)
	✓	SLA	67	23	NØRE+1		
div (ind) ✓	DIII ✓	~~CALL~~ CALL	1	~~(3)~~ 21	AREL+1 ~~NØRE+1~~ ~~5~~	1	IACC/n(ind)
Add (reverse)	AIIR = AII ✓					1	n + IACC
Subt (reverse)	SIIR ✓	SBX	6	0	1	2	n − IACC
		NGX	6	6	NØRE+1		
mpy (r)	MIIR = MII					2	n * IACC
div (r)	✓	~~LDX~~	~~4~~	~~6~~	~~NØRE+1~~		
		~~LDX~~	~~6~~	~~0~~			
		LDN	3	0	~~1~~ 1	2	n/IACC
		~~DVS~~	~~5~~	~~4~~			
		CALL	1	23	AREL+1 ~~NØRE+1~~ ~~5~~		
Add(r, i)	AIIIR = AIII					1	n(r,i) + IACC
Subt(r, i)	SIIIR ✓	SBX	6	0(3)	NØRE+1	2	n(r,i) − Iacc
		NGX	6	6	NØRE+1		
mpy(r, i)	MIIIR = MIII					2	n(r,i) * Iacc
div (r, i)		~~LDX~~	~~4~~	~~6~~	~~NØRE+1~~		
		~~LDX~~	~~6~~	~~0(3)~~	~~NØRE+1~~	~~3~~	n(r,i)/Iacc
	✓	~~DVS~~	~~5~~	~~4~~	~~5~~ ~~NØRE+1~~	1	
		CALL	1	23	AREL+1		

Standard Binary Ops (REAL – REAL)

Add	ARR	FAD	0	0	Rel=1	1	RACC + n
Subt	SRR	FSB	0	0	1	1	RACC − n
mpy	MRR	FMPY	0	0	1	1	RACC * n
divide	DRR	FDVD	0	0	1	1	RACC / n
Add (ind)	ARRI	FAD	0	0(3)	NØRE+1	1	Racc + n(i)
Subt (i)	SRRI	FSB	0	0(3)	NØRE+1	1	Racc − n(i)
mpy (i)	MRRI	FMPY	0	0(3)	NØRE+1	1	Racc * n(i)
divide	DRRI	FDVD	0	0(3)	NØRE+1	1	Racc / n(i)
Add (r)	ARRR = ARR					1	n + Racc
Subt (r)	SRRR	~~SFP~~	~~0~~	~~6~~	~~NØRE+1~~		
		~~LFP~~	~~0~~	~~0~~	~~1~~	1	n − Racc
		FSB	4	0	1		
Mpy (r)	MRRR = MRR					1	n * Racc
div (r)	DRRR	~~SFP~~	~~0~~	~~6~~	~~NØRE+1~~		
		~~LFP~~	~~0~~	~~0~~	~~1~~	1	n / Racc
		FDVD	4	0	1		
Add (i,r)	ARRIR = ARRI					1	n(i,r) + Racc
Subt (i,r)	SRRIR	~~SFP~~	~~0~~	~~6~~	~~NØRE+1~~		
		~~LFP~~	~~0~~	~~0(3)~~	~~NØRE+1~~	1	n(i,r) − Racc
		FSB	4	0(3)	NØRE+1		
Mpy (i,r)	MRRIR = MRRI					1	n(i,r) * Racc
div (i,r)	DRRIR	~~SFP~~	~~0~~	~~6~~	~~NØRE+1~~		
		~~LFP~~	~~0~~	~~0(3)~~	~~NØRE+1~~	1	n(i,r) / Racc
		FDVD	4	0(3)	NØRE+1		

Standard Binary Ops. cont.

Float Acc	SBFX ~~ELX~~	CALL	~~1~~	1~~5~~	AREL+1 ~~5~~	1	~~R~~Acc → Iacc
Float n	SBFN	100	3	0	1	2	n → %LIB (XB)
Float n (Indirect)	SBFNI	CALL	~~1~~	1~~6~~	~~5~~ AREL+1	1	n (ind.)
~~Float N~~		~~100~~ ~~CALL~~	~~3~~ ~~1~~	~~0~~ ~~14~~	~~1~~ ~~5~~	~~2~~	
~~Fix Acc~~	~~EQFXX~~	~~CALL~~	~~1~~	~~18~~	~~5~~	~~1~~	

~~Fix Acc~~

070 1

|001|011|00| 92/0

Double Precision Binary Operations (1)

Add	ADD	LDN	3	0	1	3	DAcc + n
	ADDI	1/DPAR1			0	2	DACC + n(ind)
		CALL	1	0	SPRE+1		
Subt	SDD	LDN	3	0	1	3	DAcc − n
	SDDI	1/DPAR1			0	2	DACC − n(ind)
		CALL	1	2	SPRE+1		
mpy	MDD	LDN	3	0	1	3	DAcc − n
	MDDI	1/DPAR1			0	2	DAcc * n(ind)
		CALL	1	1	SPRE+1		
	DDD	LDN	3	0	1	3	DAcc/n
div.	DDDI	1/DPAR1			0	2	DAcc/n (ind)
		CALL	1	5	SPRE+1		

~~Add(ind) ADDI~~ ~~1/DPAR1~~

Double Precision Binary Operation ②

Add (reverse)	ADDR = ADD						n + Dacc
Subt. (r)	SDDR	LDN	3	0	1	3	n - Dacc
	SDDIR	1/DPAR1			0	2	n(i) - Dacc
		CALL	1	3	SPRE+1		
mpy (r)	MDDR = MDD						n * Dacc
divide (r)	DDDR	LDN	3	0	1	3	n/ Dacc
	DDDIR	1/DPAR1			0	2	n(i) / Dacc
		CALL	1	6	SPRE+1		
Add (i, r)	ADDIR = ADDI						n(i) + Dacc
Mpy (i, r)	MDDIR = MDDI						n(i) * Dacc.

		LDN	4	0	NØRE+1
		LDN	5	0	NØRE+1
	Real → D.P. (acc)	1/DPAR1		5	2
	DPFX	CALL	1	102	SPRE+1
		LDN	3	0	1
DPFN	Real → D.P. (n)	1/DPAR1		0	3
DPFNI	Real → D.P. (n-indirect)	CALL	1	7	SPRE+1 2
DPFXX	D.P. → Real (Acc)	1/DPAR1		0	2
		CALL	1	12	SPRE+1

Not here

Complex Binary ①

Add	ACC	LDN	3	0	1	3
	ACCI	1/CPAR1			0	
		CALL	1	0	SPRE+1	2
Subt	SCC	LDN	3	0	1	3
	SCCI	1/CPAR1			0	
		CALL	1	2	SPRE+1	2
Mpy	MCC	LDN	3	0	1	3
	MCCI	1/CPAR1			0	2
		CALL	1	1	SPRE+1	
Div	DCC	LDN	3	0	1	3
	DCCI	1/CPAR1			0	
		CALL	1	5	SPRE+1	2

Complex Binary ②

Add (rev)	ACCR = ACC , ACCIR = ACCI					
Subt (rev)	SCCR	LDN	3	0	1	3
	SCCIR	1/CPAR1			0	2
		CALL	1	3	SPRE+1	
mpy (rev)	MCCR = MCC , MCCIR = MCCI					
Div (rev)	DCCR	LDN	3	0	1	3
	DCCIR	1/CPAR1			0	2
		CALL	1	6	SPRE+1	
Real → Complex (Acc)		LDN	40		NØRE+1	2
CPFX		LDN	50		NØRE+1	
		~~1/CPAR1~~				
		~~CALL~~	~~1~~	~~10~~	~~SPRE+1~~	
CPFN Real → Complex (n)		LDN	3	0	1	3
CPFNI " " (n ind)		1/CPAR1			0	2
		CALL	1	7	SPRE+1	

Small Integer Binary 1 of 1

Add	AIS	ADN	6	0	1
Sub	SIS	SBN	6	0	1
					NØRE+1
mpy	MIS	LDN	3	0	1
		MPY	6	3	NØRE+1
		SLA	67	23	NØRE+1
div.	DIS	LDN	3	0	1
		CALL	1	22	AREL+1
Add (rev)	AISR = AIS				
Sub (rev)	SISR	~~SBN~~	~~6~~	~~0~~	~~1~~
		NGX	6	6	NØRE+1
mpy (rev)	MISR = MIS				
Div (rev)	DISR	LDN	3	0	1
		CALL	1	24	AREL+1
Float small Integer		LDN	3	0	1
IBFN		CALL	1	17	AREL+1

NWRD+~~4~~=0
REL+4=AREL

Standard Exponentiation ①

Int**Int.	→ EII	LDN	3	0	1	
(indirect)	→ EIII	CALL	1	8	~~5~~ AREL+1	
Int**Int(rev)	EIIR →	LDN	3	0	1	
(rev, indirect)	EIIIR →	CALL	1	~~4~~ 12	~~5~~ AREL+1	
Real ** Int	ERI →	LDN	3	0	1	
(indir)	ERII →	CALL	1	~~4~~ 10	~~5~~ AREL+1	
Real ** Int(rev)	EIRR	LDN	3	0	1	
	EIRIR	CALL	~~1~~	~~6~~ 14	~~5~~ AREL+1	
Real ** Real	ERR	LDN	3	0	1	FEXF 4H%FEX
(ind)	ERRI	1/~~F~~FEXF			0	
		CALL	1	0	SPRE+1	
Real ** Real(rev)	ERRR	LDN	3	0	1	%FEX
(ind)	ERRIR	1/FEXF			0	0 - X**Y
		CALL	1	1	SPRE+1	1 Y**X

Standard Exponentiation (2)

Double Precision ** Real	EDR	LDN 3 0	1	FDRE 4H%FDR
(ind)	EDRI	1/FDRE	0	
		CALL 1 0	SPRE+1	
Double Precision ** Real (rev)	ERDR	LDN 3 0	1	%FDR
(ind)	ERDIR	1/FDRE	0	0 - ~~DX ** R~~
		CALL 1 ϕ	SPRE+1	1 ~~DX ** X~~ (rev)
D.P ** DP	EDD	LDN 3 0	1	
(ind)	EDDI	1/FDDE	0	FDDE 4H%FDD
		CALL 1 0	SPRE+1	
DP ** DP (rev)	EDDR	LDN 3 0	1	%FDD
(ind)	EDDIR	1/FDDE	0	0 X**Y
		CALL 1 1	SPRE+1	1 Y**X
DP ** Int.	EDI	LDN 3 0	1	
(ind)	EDII	1/DPAR2	0	
		CALL 1 0	SPRE+1	
DP ** Int (rev)	EIDR	LDN 3 0	1	
(ind)	EIDIR	1/DPAR2	0	
		CALL 1 2	SPRE+1	

Standard Exponentiation ③

Complex ** Int	ECI LDN 3 0	1
(ind)	ECII 1/CPAR2	0
	CALL 1 0	SPRE+1
Complex ** Int (no)	EICR LDN 3 0	1
ind	EICIR 1/CPAR2	0
	CALL 1 2	SPRE+1
Float Acc EFX	~~CALL~~ CALL 1 15	~~4~~ AREL+1
Float J EFN	LDN 3 0	1
Float J (ind) EFNI	CALL 1 16	~~4~~ AREL+1
R→DP (Acc) EDX	~~1/DPAR1~~ LDN 4 0	~~0~~ NØREE+1
	~~CALL 1 12~~ LDN 3 0	~~SPRE+1~~ NØRE+1
R→DP (n) EDN	LDN 3 0	1
R→DP (n ind) EDNI	1/DPAR1	0
	CALL 1 ~~4~~ 7	SPRE+1

Small Integer Exponentiation

Int ** N	EIS	LDN	3	0	1
		CALL	1	9	$ ARELH1
~~Int~~ N ** ~~N (rev)~~ Int)	EISR	LDN	3	0	1
		CALL	1	~~$~~ 13	~~$~~ ARELH1
Real ** N	ERS	LDN	3	0	1
		CALL	1	~~$~~ 11	~~$~~ ARELH1
~~Real ** N (rev)~~	~~ERSR~~	~~LDN~~	~~3~~	~~0~~	
		~~CALL~~	~~1~~		
D.P ** N	EDS	LDN	3	0	1
		1/DPAR2			0
		CALL	1	~~$~~ 1	SPREH1
Cplx ** N	ECS	LDN	3	0	1
		1/CPAR2			0
		CALL	1	~~$~~ 1	SPREH1
Float N	SEFN	LDN	3	0	1
		CALL	1	14	~~$~~ ARELH1

g) Load Accumulator

No.	Type	Mnemonic	Inst	X	Address	Rel.	No. of Inst.
✓ 0.	Int.	LI	LDX	6	0	1	1
1 ✓	Int (ind)	LII	LDX	2	0	1	2
			LDX	6	0(2)	NØRE+1	
2 ✓	Real	LR	LFP	0	0	1	1
3 ✓	Real (ind.)	LRI	LDX	2	0	1	2
			LFP	0	0(2)	NØRE+1	
4 ✓	D.P.	LD ✓	LFP	0	0	1	3
			LDX	4	2	1	
			LDX	5	3	1	
5 ✓	D.P. (ind)	LDI ✓	LDX	2	0	1	4
			LFP	0	0(2)	NØRE+1	
			LDX	4	2(2)	NØRE+1	
			LDX	5	3(2)	NØRE+1	
6 ✓	Complex	LD=LR LC=LD ✓					3
7 ✓	Comp (ind)	LCI=LDI ✓					4
8 ✓	Log	LL=LI ✓					1
9 ✓	Log (ind)	LLI=LII ✓					2
10	S.I	LS	LDN	6	0	1	1
11	Int (J)	LI3	LDX	6	0(3)	NØRE+1	1
12	Real (Y)	LR3	LFP	0	0(3)	NØRE+1	1
13	D.P (D)	LD3	LFP	0	0(3)	NØRE+1	3
			LDX	4	2(3)	NØRE+1	
			LDX	5	3(3)	NØRE+1	
14	Cplx (C)	LC3=LD3					3
15	Log (L)	LL3=LI3					1

h.) Store Accumulator

0	Int	STIW	STØ	6	0	1
1	Real	STRW	SFP	0	0	1
2	D.P	STDW	SFP	0	0	1
			STØ	4	2	1
			STØ	5	3	1
3	Cplx	STCW = STDW				
4	Log	STLW = STIW				

i) Negate

0	Int.	NI	NGX	6	6	NØRE+1	1
1	Real	NR	SFP	1	4	NØRE+1	2
			FSB	0	4	NØRE+1	
2	D.P	ND	~~CAL~~	1 / DPAR1		0	2
			CALL	1	4	SPRE+1	
3	Cplx	NC		1 / CPAR1		0	2
			CALL	1	4	SPRE+1	

j. Load X6.

LX6R	SFP	0	6	NØRE+1	1

LX6D = LX6R
LX6C = LX6R

K. Relational Binary

.LT.	RBLT	ANDX	6	0	2	1
.LE.	RBLE=RBLT	ANDX	6	0	2	2
		ERX	6	0	2	
.EQ.	RBEQ	TXU	6	0	1	4
		LDN	6	0	NØRE+1	
		SRC	6	1	NØRE+1	
		ERX	6	0	2	
.NE.	RBNE=RBEQ	X			X	3
.GT.	RBGT=RBLE				X	1
.GE.	RBGE=RBLT	X			X	2

l) <u>Logical Binary</u>

.AND.	LBAND	ANDX	6	0	1	1
.AND. (i)	LBAI	ANDX	6	0(3)	NØRE+1	1
.ØR.	LBØR	ØRX	6	0	1	1
	LBØRI	ØRX	6	0(3)	NØRE+1	1

Equals & ASF

```
EQII        STØ     6      0        1
#DEFINE     EQLL = EQII
EQIII       STØ     6      0(3)     NØRE+1
#DEFINE     EQLIL = EQIII
EQIR        CALL    1      18       AREL+1
#DEFINE     EQIIR = EQIR
#DEFINE     ASFIR = EQIR
#DEFINE     EQID = EQIR
#DEFINE     ASFID = EQIR
#DEFINE     EQIID = EQIR
EQRI        CALL    1      15       AREL+1
#DEFINE     EQRII = EQRI
#DEFINE     ASFRI = EQRI
#DEFINE     EQDI = EQRI
#DEFINE     ASFDI = EQRI
#DEFINE     EQDII = EQRI
EQRR        SFP     0      0        1
            STØ     4      2        1
            STØ     5      3        1
#DEFINE     EQDD = EQRR
#DEFINE     EQCC = EQRR
#DEFINE     EQRD = EQRR
EQRIR       SFP     0      0(3)     NØRE+1
            STØ     4      2(3)     NØRE+1
            STØ     5      3(3)     NØRE+1
#DEFINE     EQDID = EQRIR
#DEFINE     EQCIC = EQRIR
#DEFINE     EQRID = EQRIR
```

Equals & ASF

```
EQDR        LDN    4    0    NØRE+1
            LDN    5    0    NØRE+1
#DEFINE     EQDIR = EQDR
#DEFINE     ASFDR = EQDR
```

~~EQDIR~~