A Decimal Floating-Point Specification

http://www2.hursley.ibm.com/decimal/

Eric M. Schwarz

Michael F. Cowlishaw (IBM UK),

Ronald M. Smith, Charles F. Webb

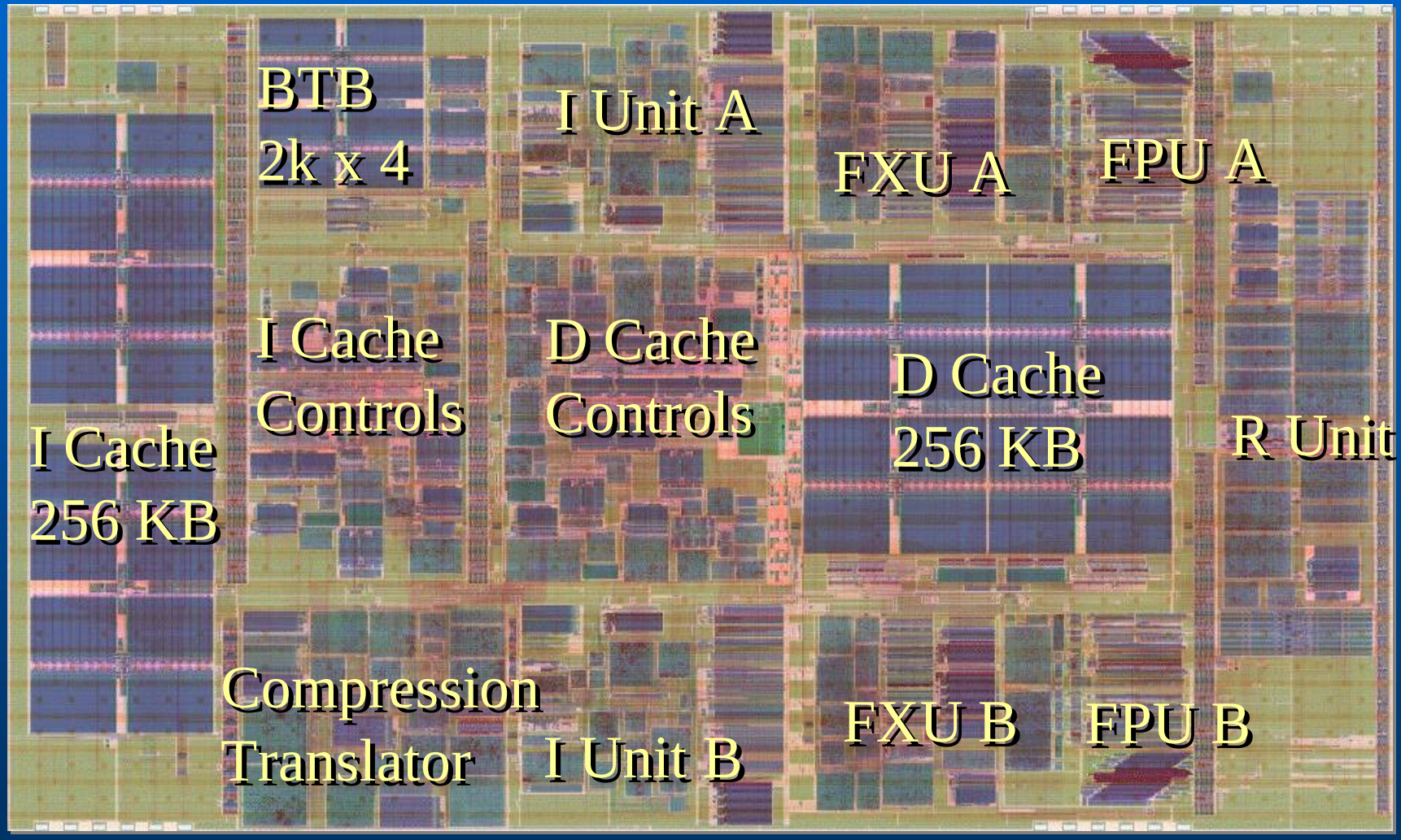
eServer Microprocessor Development

IBM Corporation

Motivation

- People think in terms of decimals
- BFP can’t represent decimal fractions such as 0.1
- Scaling of BFP requires rounding
- Decimal data is common, of all numeric data from commercial databases – 55% decimal, 26.4% small int, 17.3% integer, and 1.4% binary float
- How do we get decimal arithmetic into hardware?

z900 Microprocessor



17.9mm x 9.9mm – 47 million transistors

IBM

Goals of Decimal Floating Point Specification

- **It should allow efficient hardware or software implementation**
- **It should support numbers to be used in ANSI X3.274-1996 (REXX)**
- **It should support numbers to used in ANSI/IEEE 854-1987**
- **It should allow efficient use of existing databases**
- **It should allow for future expansion**

Software Requirements

C & C++	S/390 AS/400 Others	Fixed	31 31 38	C only C only In libraries
COBOL	All	Fixed	31	32 dig Float
C#	All	Floating	28	
Java	All	Floating	Infinite	BigDecimal
OS/400 CL	AS/400	Fixed	15	Scale <= 9
PL/I	S/390 AS/400 Others	Fixed	15 15 31	Decimal Float is actually binary
PSM	All	Fixed	31	Impl in C
Rexx	All	Floating	Infinite	
RPG	AS/400	Fixed	30	Scale <= 9

Decimal Arithmetic Specification

- Operations – add, subtract, plus, minus, multiply, divide, power, divide-integer, remainder, compare
- Format – scientific, engineering, numeric
- Rounding modes – half up, half down, half-even
- Precision control
- Flags - inexact
- Machine (concrete) representation

Machine Representation

- Sign
- Integer
- Exponent
- (-1)^sign * (integer) * 10^exponent
- Common data lengths are 4B, 8B, 10B, 16B - choose 8B and 16B.

Integer Representation

- **Binary format**
  - **Compact**
  - **Fast multiplication and division**
  - **Scaling, normalization, threshold detection slow**
  - **Requires conversion from stored BCD data**
- **Binary Coded Decimal**
  - **No conversion needed**
  - **Rounding, normalization easy**
  - **Moderately slower arithmetic than binary**
  - **Wasteful, uses only 62.5% space**
- **Compressed BCD**
  - **Minor conversion stage needed**
  - **Advantages of BCD for scaling and of Binary for compactness**

Compressed BCD Format

- **10 binary bits for 3 decimal digits**
- **7 bits for 2 digits**

Decimal Digits -> binary bits

1 2 3

Abcd efgh ijkl -> p qrs tuv wxy

Compression example

Abcd efgh ijkl -> p qrs tuv wxy

aei	P qrs tuv wxy
000	0 bcd fgh jkl
100	1 00d fgh jkl
010	1 01d bch jkl
001	1 10d fgh bcl
011	1 11d 00h bcl
101	1 11d 01h fgl
110	1 11d 10h jkl
111	1 11d 11h 00l

734d = 0111 0011 0100
-> 0 111 011 100

Compression logic

Compress 12 bit BCD to 10 bit binary encoding:

p = a | e | i

q = (b e') | i | (a e)

r = (c i') | e | (a i)

s = d

t = (a e) | (f (a' | i')) | (b e i')

u = (a i) | (c e i') | (g e')

v = h

w = j | (b i) | (f a i)

x = k | (c i) | (g a i)

y = l

Expansion Logic

Expand 10 bit binary encoding to 12 bit BCD:

a = (p q' r') | (p q r (t | u))
b = (q p') | (t p q' r) | (w q (r' | (t' u')))
c = (r (p' | (q' u))) | (x p q (r' | (t' u')))
d = s
e = (p r (q' | u' | t))
f = (t (p' | r')) | (p q r t' u w)
g = (u (p' | r')) | (p q r t' u x)
h = v
i = (p q (r' | t' | u))
j = (w (p' | q' | (r t)))
k = (x (p' | q' | (r t)))
l = y

Exponent Representation

- **Binary two's complement**
  - **Allows for fast addition**
  - **Compact**
- **Binary unsigned with bias**
  - **Similar advantage to above**
  - **Similar to IEEE BFP**
- **BCD**
  - **No conversion**
  - **Inefficient and slow**

Special Values

- **Reserved Exponent Values**
  - **Similar to IEEE BFP**
  - **Compact**
- **Using separate bits**
  - **Easier to detect**
  - **wasteful**

Length of Exponent

- **Must meet IEEE 854 standard**
  - **(Emax – Emin) > 5 x Digits prefer 10 x D**
  - **Double precision 8 x Emax single + 7**
  - **Recommend Emin = - Emax**
- **Short (8B) – 11 bit (-999 to +999-(D-1))**
- **Long(16B) – 15 bit (-9999 to +9999-(D-1))**

Decimal Digits Possible

	BCD	7:2	10:3	Binary
Single (52 bits)	13 digits (0)	14 digits (3)	15 digits (2)	15 digits (2)
Double (112 bits)	28 digits (0)	32 digits (0)	33 digits (2)	33 digits (2)

Proposed Decimal Format

Sign	Exponent	Filler	Integer
1 bit	11 bit unsigned binary w/bias	2 bit	15 digit (Cmpr BCD 10:3)
1 bit	15 bit unsigned binary w/bias	2 bit	33 digit (Cmpr BCD 10:3)

Remarks

- Determined a possible machine representation
- Would like multiple manufactures to support a representation
- Need to keep software standards inline with hardware limitations
- Encourage hardware implementation of decimal floating point

- http://www2.hursley.ibm.com/decimal/