Cairo University
Electronics and Communications Department

Computer Arithmetic: Decimal addition and multiplication

Hossam A. H. Fahmy

What is different in DFP?

Decimal formats

- may be in either binary integer decimal (BID) or densely packed decimal (DPD) encoding,

  and

- may have leading zeros.

What are the changes due to these facts?

BID encoding

The combination field determines the most significant bits of the significand. Those and the trailing significand are treated as one integer in unsigned binary notation.

- If there is no need for exponent adjustments, then an operation on two such significands is very easy. It may use the existing binary hardware.
- When exponent adjustments are necessary, the adjustment does not lead to a simple shift of the significand. We must multiply (or divide) by $10^{\text{exp\_diff}}$.
- Since the "digit boundaries" are not directly available, rounding is not straightforward.

Use a multiplier to add!

To align the significands for addition, *multiply* by $10^{\text{exp\_diff}}$.

- Use the difference of exponents to address a table and lookup for the binary representation of $10^{\text{exp\_diff}}$.
- Use a binary multiplier to multiply it by one operand. (May be the larger if you enlarge the datapath to the left.)
- Add or subtract the other operand depending on effective operation.
- Check for negative results if the exponents were equal.

Rounding

1. Count the number of digits in the intermediate result.
   - May be done by a *lookup table* depending on the bit position of the leading one.
   - A correction is needed in some cases: detected by comparing the number to powers of 10 saved in a *lookup table*.
2. To round off the least significant d digits, either
   - divide by 10^d then check the remainder (long time)

     or
   - *multiply* by 10^{-d} then check the remainder (large area).

Reuse the multiplier

- For the multiplication operation, BID significands are simply "binary integers".
- The rounding is still more difficult than BFP as in the case of DFP addition.

$\Rightarrow$ A single 64×64 bits multiplier may be reused for BFP, alignment for add/sub in BID decimal64, and rounding for the various BID decimal64 operations.

DPD encoding

Each ten bits of the trailing significand encode three decimal digits. The conversion from a 'declet' to three separate BCD digits and the conversion back is easy.

The accessibility of the digit boundaries simplifies

- the alignment of operands via simple shifts by any number of digits,
- the rounding of the result at the exact decimal digit boundary required by the standard,

  and
- the conversion from or to character strings for inspection by humans.

Life is not rosy with DPD

- DPD FP addition uses BCD not binary adders and must handle the leading zeros correctly. Beyond that, it is similar to BFP.
- Multiplication is much harder:
  - Each digit in the multiplier may be from 0 to 9 $\Rightarrow$ many multiples of the multiplicand.
  - Addition of conventional BCD (8421) needs correction $\Rightarrow$ no simple carry-save partial product reduction scheme.

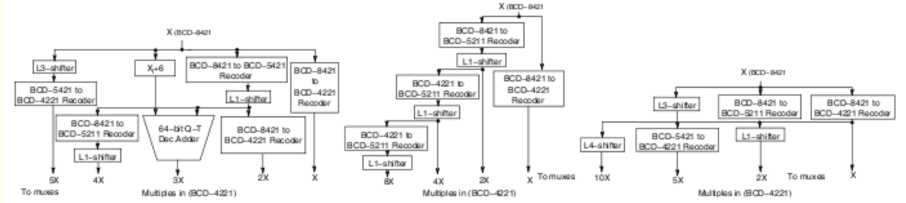
Multiplier recoding in DPD

- The direct use of the multiplier digits in BCD-8421 leads to complex multiples of the multiplicand (3X, 6X, 7X, 9X).
- A ‘modified Booth’ recoding leads to a digit set $\{-5, \ldots, 5\}$ with multiples 0, X, 2X, 3X, 4X, 5X, and their negatives.
- Other recodings were proposed such as
  - $Y_i = 4Y_i^U + Y_i^L$ with $Y_i^U \in \{0, 1, 2\}$ and $Y_i^L \in \{-2, -1, 0, 1, 2\}$ requiring 0, X, 2X, and their negatives as well as 4X and 8X

    or
  - $Y_i = 5Y_i^U + Y_i^L$ with $Y_i^U \in \{0, 1\}$ and $Y_i^L \in \{-2, -1, 0, 1, 2\}$ requiring 0, X, 2X, and their negatives as well as 5X.

Multiplicand multiples

Remember that a left shift by one bit of BCD-5211 generates 2X encoded in BCD-4221. Similarly, a left shift by three bits of BCD-8421 generates 5X encoded in BCD-5421.



(a) Multiples for SD radix–10 encoding. (b) Multiples for SD radix-4. (c) Multiples for SD radix-5.

(From Vazquez *et al.* in 18th Arith Symposium)

Decimal $(3, 2)$ counters

		Decimal Value	BCD 4221	
A :		3	0101	Binary
B :	+	7	1101	full adders
C :	+	8	1110	
S :		4	0110	
$H4221$:		7	1101	
				Recoding $BCD4221$ to $BCD5211$
$H5211$:		7	1100	
				Left Shift
$W = 2 \times H$:		14	1100-	W in $BCD4221$

Result $= S + 2H = S + W = 18$

Division and square root

- Once addition and multiplication are done, the other functions are “straightforward”.
- Some used digit recurrence techniques (SRT) and some used Newton-Raphson.
- Careful error analysis is needed and the rounding must be correct.

Energy

The use of hardware to perform decimal operations instead of software leads to

- a much shorter time to finish the operation (factor may be 100 to 1 or more)

  and

- no energy consumed in overheads such as fetching and decoding of software instructions.

However, the additional circuits consume static power when idle and burn energy.

In general, the use of decimal HW is much more energy efficient than SW.