Royal Digital I Calculator

Theory of Operation

bhilpert / madrona.ca
2020 May
Rendition: 2022-Mar-31

Contents

Introduction

The Royal Digital I is a basic 4-function calculator from the early 1970s. One unit of this model was obtained, examined, and reverse-engineered to produce a schematic and this commentary. The unit examined contains components with date codes from 1971.

This commentary describes the functioning of the logic of the Digital I. This commentary is derived from, and intended as an adjunct to, the reverse-engineered schematic. The schematic document includes a block diagram, state diagram, and procedure state-sequence examples.

Text within square brackets refers to the logic element (flip-flop, gate, gate input, etc.) which implements or controls the function described in the associated text. Discrete gates are identified by referring to one of the inputs with a "G" prefixed to the input identifier, e.g. GD110 refers to the gate which has an input formed from D110 and may be found in the schematic by searching for "D110". Logic elements in ICs are distinguished with a suffix of an input or output pin, e.g. UB1.13, which may be found in the schematic by searching for "UB1" and looking for pin 13.

Architectural Overview

The Digital I is a bit-serial design. Data processing involves two 48-bit shift registers, each holding a 10-digit decimal number, and a serial BCD arithmetic unit. Data is continually rotating through the registers, for processing, and for supplying the multiplexed display during idle.

A state machine directs the logic through higher-level procedures to implement number entry and the 4 basic arithmetic operations.

The Digital I model comes at the end of the dedicated-hardware, bit-serial era of calculator implementations. The logic elements and the algorithms of the Digital I present a very finely tuned and optimised design for a bit-serial calculator, in comparison to typically more complex and clunkier designs from earlier in the era.

Physical Implementation

The Digital I is implemented using ICs from the 'JMOS' (Japanese MOS) family of SSI/MSI PMOS ICs. Use of this family of ICs typically involves many gates constructed from discrete diodes and resistors. Active elements are PMOS transistors within the ICs while most gate elements are discrete diode logic. Both AND and OR gates are formed from discrete diodes.

Documentation for several of the IC types used was not previously available, either as original datasheets or from earlier reverse-engineerings of other calculator models. These ICs are: μPD107C, M5811, M5812, M58214, M58221. Their pinout and internal operation have been derived from their instances in this calculator. Pinouts and functions of the other ICs were derived during reverse engineering of other calculator models.

Here again, at the hardware level, the design is heavily optimised, to reduce component count. Notably, for many discrete gates the load resistors are used as an input rather than being taken to a fixed supply voltage, thus eliminating a diode for each gate so treated.

Logic Presentation

The design uses a combination of positive and negative logic. The data bitstreams and logic are primarily positive logic, the control system is primarily negative logic.

In this commentary:

1 = H = high = 0V

0 = L = low = –24V

When a signal is said to be *asserted*, the signal may be either 1 or 0, depending upon whether it is in positive logic or negative logic respectively. E.g. if DPC is asserted, DPC=1 and (DPC~)=0.

Negative logic gates are shown in the schematic in the appropriate complementary boolean form.

OEM TEAL and the abc-800

The Royal Digital I was OEM'd by Tokyo Electronics Application Laboratory (TEAL). TEAL also OEM'd a 12-digit Nixie-display calculator marketed as the abc-800. The logic designs of the Royal Digital I and abc-800 are nearly identical. Most likely the abc-800 slightly preceded the Digital I with the abc-800 design being adapted for the Digital I. Some noted differences are:

- The AU in the abc-800 is implemented in the usual JMOS SSI/discrete manner. This AU design is identical to that in another model OEM'd by TEAL marketed under the Riccar name. The AU in the Digital I is contained in an MSI IC.
- Alterations to the DPC logic.
- Minor changes to a few Next-State gates.
- Minor changes to some condition gates setting the Error Flag.
- The display, decoder/drivers and power supply, of course.

Even though the Digital I has two fewer digits than the abc-800, the registers are the same length.

Given the similarities between the two models, this commentary is also largely applicable to the abc-800.

Architectural Elements

Timing Generation

Clocking for the logic can be discerned into 3 levels: bit timing, digit timing, and number/state timing.

- Master Clock

  Timing begins with the master clock, a potted module generating two non-overlapping active-low clock phases. One of these phases is the capture pulse (ØC), during the low period of ØC logic states are captured by flip-flops. The other phase is the update pulse (ØT), flip-flop output transitions occur on the negative edge of the ØT pulse.

  Bit-shifting of the serial bit streams is performed by these pulses.

- Bit-of-Digit Phases

  4 cycles of the master clock form the timing for a 4-bit BCD digit. A 3 flip-flop state machine [UH1] forms a counter to produce 4 phases: ØB1, ØB2, ØB4, ØB8.

- Digit-of-Number Phases

  13 digit cycles define the timing for a full rotation of the number registers. A ring counter [UH2,3] distinguishes these cycles into distinct phases, ØD0::ØD12.

  The ring counter is of standard design although this is not immediately apparent as most of the ring counter logic is in the ICs. The ICs each contain a 6-bit shift register, these are chained into a 12-bit shift register. The ICs also contain a gate which detects when all the FFs of the IC are clear [ACOUT], these indications are combined [GD16]. When any and all 1s in the shift register have been shifted out, all FFs are 0, the ACOUTs go high and a 1 is injected into the first stage [UH3.D1]. The 13th digit-phase arises from the period between the last 1 being shifted out and the next clock registering the injected 1. This phase is ØD12.

- Number Cycle

  The 13 digit-phases thus define a number cycle. A capture pulse (ØEON) is generated at the end of a number cycle by the coincidence of ØD12, ØB8 & ØC [GD74].

RX and RY Registers

The RX and RY registers are 12-digit (48-bit) shift registers contained in one IC [UD4].

- Encoding — The data of each register represent a 10-digit decimal number, each decimal digit encoded in standard 4-bit 1248 BCD. Each register is looped, with constantly shifting, recirculating data. In time, the data is ordered LSB of LSD first. The digits may be enumerated from 0::11, e.g. RX[0] is the LSD of RX. Digit n emerges from a register during ØDn.
- Extra Digits — The two extra digits of each register are an inheritance from the 12-digit abc-800 design. In the Digital I, these 'hidden digits' do alleviate some anomalies present in the abc-800, such as not being allowed to have division operands fill the display MSD as it must be kept available for the dividend being shifted up after overdraft.
- Data Paths — The X and Y registers may loop on themselves, in which case the data loop is 12 digits long, or, either or both may be directed to loop through the Arithmetic Unit, which adds one digit (4 bits) to the loop length, the data loop for the register becomes 13 digits long.
- Clocking — The full number cycle is 13 digits long, thus if a register is looped on itself, the bit shift clock must be suppressed for one digit-phase to maintain the shift register contents at a fixed position across number cycles. Clocking of the registers during ØD12 may be suppressed to this end.
- Shift down — If a register is clocked for one digit-phase more than the loop length, e.g. the register is clocked for 13 digit-phases when the data loop length is 12 digits (12D/13Ø), the data is moved to an earlier (lower) digit-phase, and the number is rotated right or shifted down one digit (÷10).
- Shift up — If a register is clocked for one digit-phase less than the loop length, e.g. the register is clocked for only 12 digit-phases when the data loop length is 13 digits (13D/12Ø), the data is held back while time moves on, the number is rotated left or shifted up one digit (•10).

AU - Arithmetic Unit

The arithmetic-performing logic for the calculator is contained in one IC [UE4]: a serial BCD adder/subtracter. This IC would contain a serial binary adder including a carry flip-flop, a 4-bit shift register, tens-carry logic, and BCD sum correction logic.

- Carry Suppress — In a serial adder, carry is propagated in time from bit-to-bit, and for BCD operation, ten's carry must be propagated from digit-to-digit. Such facility is normally in effect in the AU. An input on the AU IC [UE4.6] permits disabling carry propagation. This is used during division to suppress borrow into RX[11].
- N8,4,2,1 — The 4 bits of the AU are available in parallel. They are used to feed the display in multiplex and to detect a zeroed digit.

While the specific internal logic organisation of the AU IC is not available, a functional-equivalent logic diagram is presented in the schematic. The following describes the operation of that functional-equivalent logic, to give an idea of what must be accomplished and one technique for doing so.

- The problem — The binary sum of two 4-bit BCD digits may be greater than 10, necessitating correction to normalised BCD form. For example 8 (1000) + 5 (0101) = 13 (1101). The binary sum of 1101 must be corrected to a BCD-digit sum of 3 (0011) and a carry to the next BCD digit.
- A solution — Observing that 4 bits is modulo-16, and 16 is 6 greater than 10, adding 6 to the binary sum will give the correct BCD-digit sum: 13 (1101) + 6 (0110) = 19 (10011). The lower 4 bits of 10011 is 3.
- Binary adder — A raw single-bit binary sum is produced from the two operand bits (A & B) and carry using two XOR gates [AB,SUM]. A binary carry is generated from several gates culminating in the gate [CRY].
- Detecting the problem — The raw binary sum is fed into 4 bits of storage [N8,N4,N2,NN1] making the digit sum available in parallel. Gates [N8+N4,TC] detect bit patterns for digit sums ≥ 10 by sampling at the end of digit, ØB8. This additional ten's carry is combined with the binary carry for latching into the serial carry flag [CF]. The carry flag is now valid as both an intra-digit bit carry and an inter-digit carry, and so is looped back as the serial carry input to the carry logic, via carry suppression gate [C]. As well, at the end of digit the carry flag indicates that sum-correction is needed.
- Generating a 6 — With correction needed, at the end of ØB1 flip-flop [SIX] is set high via gates [CORR,SIXD], and so is high for ØB2. It is looped back via [SIXD] so remains high for ØB4, but is reset on ØB8, thus producing 0110 in serial.

  While ØB1 is available externally, flip-flop [ØB1] is used internally as there is no pin for ØB1 on the IC.
- Normalising adder — The binary sum, delayed through 4 bits, and the correcting 6, are fed to another binary adder [NS6,N1] and carry logic with serial flag [NCRY,NCF,etc.]. The carry logic is simplified a little thanks to the sub-set of bit patterns seen with these operands. Inter-digit carry is suppressed here.
- Subtraction — For subtraction all that must be done is alter the carry-generation [AINV,NSINV].

Control State Machine

The Control State Machine (hereafter, the Machine) sequences the higher-level procedures of the calculator: number entry, decimal point alignment, multiplication, etc.

• Organisation — The Machine is of fairly regular and conventional organisation. Ten D-type flip-flops in two IC packages form the state registers [UC1,UD1]. The state is not binary-encoded in the flip-flops, rather the FFs indicate 1-of-10 states: ST0::ST9. Gates in a standard AND-OR organisation form the next-state logic determining state transitions, feeding the D inputs of the flip-flops.

• State Clocking — State transitions are synchronous to the number cycle – an individual state cycle processes one pass through the RX and RY registers. State transition requests asserted by the next-state logic are captured at ØEON (End of Number). The transition actually occurs with ØT. State cycles and number cycles are in essence synonymous.

• Implicit Looping — The ICs containing the state flip-flops have a feature to facilitate state loops without dedicated next-state logic. An internal gate detects whether any next-state assertions are present at the FF D inputs [RQOUT]. If there are no assertions at the end of the state cycle the FF D capture is suppressed [GD105]. The FFs thus do not change and the existent state persists for the subsequent state cycle. The state register ICs are chained [RQIN] to effect this facility across all 10 state FFs.

• Multiple States — Three gates formed by MA11 are used to suppress the expression of States 5, 6 & 8, under the conditions of MA14. Most of the conditions are to prioritise the expression of another state because there were multiple next-state assertions.

For example, when in State 7 during multiplication and PLT is asserted, next-state logic [MO7] asserts a request for State 9, which indeed is the algorithmic intention. However, other next-state logic [D110] also asserts a request for State 5. Once in State 9, the expression of State 5 is suppressed [MA14.3,GMA11.4]. The suppression is done via the diode of MA11.4 holding the line high. While this can be seen as a negative-logic AND gate with one diode and one wired input, more likely what is actually happening is the state flip-flop is being force-set by drain-triggerring via the diode.

Why these state-suppressions were done rather than setting up the ‘correct’ next-state logic has not been determined, nor why they were done with a single condition gate and control line. The effects that have been determined as useful are:

- KPN suppresses transition from State 3 to State 0.
- State 0 suppresses State 8.
- State 7 suppresses State 6.
- State 9 suppresses State 5.

Keyboard and Operation Flags

• Key requests

Three signals are generated from key-presses to invoke procedure execution by the Machine. These are:

KPN – process a numeral key
KPO – process an operation key
KPD – process the decimal point key.

These signals are synchronised to the number cycle by flip-flops [UA2].

• OP flags

As the calculator implements a form of arithmetic/infix notation rather than RPN/postfix it is necessary to register presses of the multiply and divide keys for future reference when the += key is pressed, to know which procedure to perform. Two flags, formed from SR flip-flops, act to this end. One flag distinguishes Multiply and Divide (OPMD) from Add and Subtract (OPAS). The second flag distinguishes Add and Multiply (OPAM) from Subtract and Divide (OPSD).

The OP flags do not necessarily immediately reflect the operation key pressed, rather they may be adjusted when the Machine returns to ST0 [MA6] after execution of some procedure invoked by the keypress. For example, the multiply and divide keys actually result in the addition procedure being executed (to null effect as RY=0), but when that procedure returns to State 0, the OP flags are set appropriately for multiply or divide.

• Options

There are some jumpers and diodes around the keyboard section labeled in a manner suggestive of alternatives for keyboard behaviour. These may relate to the Royal Digital I-K model.

DPC – Decimal Point Counter

The Decimal Point Counter is a 6-stage looped shift register [UA1] used to track the decimal point position for the number in RY, and as a multiplex latch for the decimal point during idle.

• Clocking — The DPC is clocked at a per-digit rate.

By default, the DPC receives 12 clocks per number cycle, the clock during ØD12 being suppressed [GD95]. The DPC then, being 6 bits, is actually rotated twice during such a number cycle.

• Marking — By default, 0s are injected into the DPC shift register. A single-bit 1-mark may be injected, to circulate through the shift register.

At an abstract level, the DPC may be considered to take 6 values:

N: Null. The DP shift register is empty (all-0s).
0,1,2,3,4: The number of decimal places for the value in RY. Physically this is indicated by the ØD phase during which the shift register input [UA1.1] is 1.

The DPC is N if the DP key has not been pressed during number entry, and, during multiply & divide operations, will be decremented to 0 and transformed to N. The All-Clear output [UA1.9] from the shift register makes this state readily available as DPC=N.

• Increment — The DPC value is incremented by shifting the mark up one digit-phase (left in the shift register, closer to the input). This is accomplished by suppressing the clock during ØD11 [GD113.100K=L] in addition to the normal ØD12 suppression, giving an 11-clock cycle.

• Decrement — The DPC value is decremented by shifting the mark down one digit-phase (right in the shift register, closer to the output). This is accomplished by enabling the normally-suppressed clock during ØD12 [GD95.100K=H], giving a 13-clock cycle.

• Nulling — In operations where the DPC is being decremented, when it hits 0, it will be transformed to N.

GD79 is normally enabled, allowing recirculation of the mark. When this path is disabled [GD79.100K=H], there is an alternative path through GD62. However this 2nd path is shut off during ØD0, suppressing reinjection of the mark when the DPC has been decremented to 0. The DPC shift register is now empty, and DPC=N is asserted.

PLT – Primary Loop Tracker

The Primary Loop Tracker is a 12-bit looped shift register [UE1], used to track iterations of the primary loop of the multiplication and division procedures. The manner in which it is used to this end is different between multiplication and division, so is discussed in the according commentary.

• Clocking — The PLT is clocked once per digit (ØB1).

By default, the clock is suppressed on ØD11 [D153], giving 12 clocks per number/state cycle. Bits circulating in the (12-bit) shift register thus stay in the same time-relative position, that is, they sit at a static digit position across number/state cycles.

• Tracking — An iteration of a primary loop is multiple number/state cycles however, so - being clocked at a much higher rate - the state of the PLT at any given instant is not going to be indicative of primary loop iterations. That is, it is not clocked merely once for each iteration. The PLT instead operates as what may be called a slipping ring counter. An additional clock or a suppressed clock cause bits to slip a position up or down respectively - they will emerge from the shift register one digit-phase earlier or later.

There are then, two notions of shifting or rotation going on. One is the low-level, one-way clock-driven rotation of the bits through the shift register flip-flops. At a higher level is the data state relative to the digit-phase of the number cycle. Here, the rotation may be either up or down. Rotating up means a given bit emerges from the shift register at a later/higher-numbered digit-phase (with loop-around from ØD11 to ØD0). Rotating down means a given bit emerges from the shift register at an earlier/lower-numbered digit-phase (with loop-around from ØD0 to ØD11).

• Marks — The clear or empty state of the PLT is all-1s. One or more single-bit 0-marks may be injected into the PLT. A 0-mark emerging from the PLT shift register asserts PLT (PLT=1).

• State 0 clear — In State 0, the PLT shift register is cleared to all-1s [GD123].

Error Flag

The Error Flag [UC4.13,12] and associated gates detect various error and overflow conditions. An error condition forces the Machine to State 0 [D197], clears RX [MA16.2], and forces the display of full-height zeroes and all 5 DPs. The Error Flag is reset by the Clear key [D87].

• Entering too many digits

In State 2, if the 10^{th} digit is non-zero and the number is not yet in alignment with the DP setpoint, too many digits are being entered.
e.g. 12345678901
[MO12]

• Overflow (upper digits)

When an operation has completed, State 0 is entered. If RX[10] or RX[11] are non-zero, overflow occurred.
e.g. 9 999 999 999 * 2
[GCD168]

• Overflow (carry)

If AU addition is being performed in State 6 and there is a 10s-carry out of RX[9], overflow occurred. ("R<0" is a misnomer here, as it is an indication of carry rather than borrow.)
e.g. 9 999 999 999 + 2
[MO13]

• Overflow (product)

Overflow for multiplication is detected by assessing whether the number of digits which may be required for the product will exceed the register capacity.
During State 6, as additions to the product are being performed, the DFLG is latched when the PLT mark emerges from the PLT [GD97]. If a carry from the AU occurs [D124] or Y is non-zero [D64] with DFLG asserted - thus above the PLT mark - then the product will overflow the allowable register capacity.
(Example for Royal not found, may only be applicable to abc-800.)
[MO11]

• Overflow (quotient multiply)

During divide, in State 5, if it is necessary to shift up due to decimal places in the divisor [MO10.100K], without empty digits at the upper end [MO10.1], overflow occurs.
e.g. 12345678.9 ÷ .01, DPS=2
[MO10]

• MSD not clear for division

During divide, the presence of non-zero numerals in RX[11] or RY[11] is sensed during State 4, as the divisor is being shifted up. For the Digital I this does not appear to be relevant as there should be no way for numerals to be entered into these digits, and State 4 is exited by sensing numerals being shifted up during the earlier ØD10. However for the 12-digit ABC-800 it is relevant, as numerals can be entered into these digits but they must remain clear to be available for shift-up after an overdraft.
[GD148]

Display

• Latch/decode — Digit data from N8,4,2,1 is captured into the display latch [UF2] with the first ØC pulse of digit-phases (ØB1~•ØC) [GD12], and decoded to segment data.

• Zero conversion — The numeral zero is detected at the output of the decoder and several segment controls altered to produce a half-height zero rather than full-height zero in the display [GD13,etc.].

• VF tubes — The display proper is 10 individual-digit vacuum-flourescent tubes and associated drivers for the segments and grids. The heaters are in a series-string and supplied with a negative voltage (V–D) at a tap in the middle of the string. The grid and segment drivers close to GND when asserted, providing a conduction path where the grid or segment is positive relative to the heaters.

The driver ICs can be expected to have pull-down resistors to V–D on each output. The resistor and capacitor between V–D and the heater string produce some voltage drop so the heater is a few volts more positive than V–D. Thus when a driver is not asserted the grid or segment is pulled more negative than the heater to ensure cut-off.

• Error Indication — If ERR is asserted, zero conversion is suppressed [D18] and the DP segment is turned on continuously [GD17]. Only 5 of the VF tubes have the DP segment connected to the DP driver. The error display is thus a display filled with full-height zeroes (RX having been cleared by asserted ERR) and 5 decimal points.

Procedures

State 0 Idle

After an operation completes the Machine enters State 0 to idle. State 0 loops implicitly until a key is pressed.

- RX display — RX is looped through the AU on a 13D/13Ø loop [UD3.13=H,D131=H,UC2.6=H]. RX being the register supplying the Nn signals, it is displayed.
- DP display — A value in RX has earlier been aligned with the switch-selected DP setpoint. For the display, the DP segment merely needs to be turned on in accordance to the switch setpoint.

  A mark is injected into the DPC at the switch-selected ØDn digit-phase [GD59] and one digit-phase later asserts DP, which controls the DP segment in the display. This one-digit delay coordinates with the display multiplexing.
- RY clear — RY is cleared by disabling its data loop [D99=L].

State 1 Idle

After a numeral key or the DP key has been processed, the Machine enters State 1 to idle. The Machine loops in State 1 until another numeral key or operation key is pressed. Numbers are entered into RY, so RY is displayed.

- RY display — RY is looped through the AU on a 13D/13Ø loop [D98=L,MA15.1=L,UC2.8=H]. RY being the register supplying the Nn signals, it is displayed.
- DP display — The DP position displayed varies with the value of the DPC. The mark circulating in the DPC asserts DP one digit-phase after the DPC mark injection, which is one more than the DPC value, e.g. if DPC=1, DP is asserted during ØD2, which coordinates with the display multiplexing.

Number Entry

• KPN assert — Pressing a numeral key asserts KPN [MA2,UA2]. KPN is synchronised to the number cycle by a FF [UA2.12]. If the Machine is in State 0, a transition request to State 1 is asserted [GD22,GD42].

• State 1 — RX is switched to loop by itself on a 12D/12Ø loop [MA13.2=H,D111=L].

RY is switched through the AU on a 13D/13Ø loop [D98=L, UD3.9=H, MA15.1=L, UD2.6=H]. With RY on a 13-digit loop, what amounts to the 13th digit is cleared during ØD0 [GD137].

The Machine proceeds to State 2 [D67].

• State 2 — The numeral is entered into RY, and RY is shifted up a digit. The DPC is incremented if warranted.

The numeral key is encoded into BCD [MA3,4,5,GD3], serialized into a bitstream [GD13,14,15,16,MA1], and injected into RY during ØD0 [GD92].

The digit-shift is accomplished by suppressing clocks to RY during ØD12 [UC2.11=L], although no digit-shift is performed if the maximum number of decimal places has already been reached [DPC<SP not asserted].

If DPC≠N and DPC<SP is asserted, the DPC mark is shifted up to increment the number of decimal places.

The Machine proceeds to State 3 [GD70].

• State 3 — A dummy state which simply loops waiting for KPN to deassert [MA14.200K, GMA11.2], to avoid multiple numeral entries. When KPN deasserts, the Machine proceeds to State 1 to idle.

• DP entry — If the DP key is pressed, KPD is asserted. The Machine may be idling in State 0 or State 1, if in State 0 it moves to State 1. DPC=N asserted indicates the DP key has not already been pressed [MO2.1]. If asserted, a mark is injected into the DPC at ØD0 [GD19,MO2], setting DPC to 0. The Machine loops in State 1.

This same facility is used to set the DPC to 0 [D21] when the Machine passes through State 1 at the behest of an operation key.

Decimal Point Alignment

With a number being entered into RY, the Machine is in State 1. Pressing an operation key, without multiply or divide having already been registered as the operation to perform, will take the Machine to State 2, to align the number to the DP-switch setpoint.

• DPC<SP Flag — The DPC<SP flag is an SR flip-flop indicating whether the number in RY is aligned to the switch-selected DP setpoint.

The DPC<SP flag is set at the beginning of each number cycle [GD77], and, when not in State 0 [GD18], is reset during the number cycle if and when the mark from the DPC shift register is synchronous [GD37] to the setpoint digit-phase.

• State 2 — If DPC<SP has not been reset [GD94] by the time of ØD11, that shift of the DPC is suppressed [GD113], incrementing the DPC, and a digit-shift of RY is suppressed [UC2.11] to shift the numeric data up one digit.

The Machine loops in State 2 until the decimal point is aligned to the setpoint. Once aligned, the Machine proceeds to State 6 [GD69].

Addition & Subtraction

State 6 is entered from State 2 to perform a simple addition or subtraction.

• Registers — RX and RY [MA15.2] are both fed to the AU, with RX on a 13D/13Ø loop [UC2.6=H] and RY also on a 13D/13Ø loop [GD98.100K=H]. The arithmetic function is performed during the number cycle and the result injected into both RX and RY.

• NEG flag — The NEG flag is a toggled flip-flop [TR1,TR2] indicating the sign of the number in RX.

• Add or subtract? — The arithmetic function performed by the AU is determined by both the keyed operation and the sign of the number in RX. If they do not match, e.g. RY is being added to a negative number in RX, SUB is asserted [MO3,MO4,GD45], the AU function is subtract. If they do match, the AU function is addition.

• Pos or neg? — If an AU subtraction results in a value less than zero, the result from the AU will be in 10s-complement form and a borrow will propagate out of the AU carry output at the beginning of ØD12 [UE4.3], thru UC3.6, to assert R<0.

The state of R<0 is latched across the digit-phase [UB2.11,UC3.8] so it can be detected at ØEON for next-state determination. If R<0 is not asserted, everything is fine and the Machine proceeds to State 0 [GD44]. If R<0, the Machine proceeds to State 8 to negate the 10s-complement result so it will display as an ordinary value.

• State 8 — AU subtraction is enabled [D50], and 0 fed to the AU A input by blocking the bitstream from RX [D129]. RY is fed to the AU B input [MA15.3]. RX receives the result. Thus: 0 – RY ==> RX.

The NEG flag is toggled to flip the displayed sign of the number in RX [D71], and the machine proceeds to State 0 [MA9.4].

Multiplication

Multiplication is performed via the conventional algorithm of repeated additions of the multiplicand to an accumulating product, those additions being limited by a decrementing digit of the multiplier, with repeated shifting of numbers to position the multiplicand appropriately to the product for the additions to occur.

• Loops — There are three loops in the algorithm, a sub-loop in which the additions and the decrements occur (State 6), a primary loop within which the sub-loop and shifting occur (States 5,6,7), and a third loop near the end to account for decimal places (State 9).

• PLT use: — The Primary Loop Tracker is used as a counter to determine a fixed number of iterations (12) through the primary loop. Each iteration will serve to process one digit of the multiplier.

With the PLT having been cleared to all-1s, a single-bit 0-mark will be injected into the PLT shift register at the beginning of the multiplication procedure, in State 1. For each iteration of the primary loop, the mark will be rotated up one digit-phase in State 5. After 12 iterations the mark will rotate back to the output end of the PLT shift register where it asserts PLT at the end of the number/state cycle. This is detected at the end of State 7, to cause the primary loop to exit.

• Preconditions — Prior to the multiplication process proper the following preconditions exist:

- the multiplier has been entered into RX and aligned to the DP setpoint,
- the multiplicand has been entered into RY,
- the DPC has been set to the number of decimal places of the multiplicand (RY),
- OPMD and OPAM have been registered by the press of the multiply key, these assert OPM.

• Product — The product will accumulate in the lower digits of RX.

• State 1 — *Mark the PLT*

The PLT having been previously cleared to all-1s, a single-bit 0-mark is injected into the PLT shift register during ØD12 [GD139].

• State 4 — *(Dummy state)*

A duplicate 0-mark is injected into the PLT [GD149]. This mark overlaps with the mark from State 1 so there is nonetheless only one bit marked in the PLT register.

• State 5 — *Shift up multiplier and product*

The multiplier and accumulating product (both in RX) are shifted up one digit. RX is placed on a 13D/12Ø cycle: RX is looped through the AU [GD132 enabled, MA13 disabled], with ØD12 suppressed [UC2.4,GD85].

The PLT mark is rotated up. The PLT clock is suppressed during ØD0 [GD154] in addition to the usual suppression during ØD11, giving an 11-digit cycle. The mark thus slips up one digit-phase.

If a non-zero digit of the multiplier has shifted into RX[11], the Machine proceeds to State 6 [D90] to perform additions of the multiplicand to the product. If RX[11]=0 [GD159,GD119], there is no need for State 6, so the Machine proceeds to State 7 to test for the end of the primary loop. This selection is an instance of the multiple-state suppression discussed in the State Machine commentary. The transition of State 5 to 6 is unconditioned, but if State 7 is entered, ST6 is suppressed [MA14.1,MA11.3].

• State 6 *Add & Decrement*

RY is added to RX during ØD0::11.

During ØD12, RX[11] is decremented, RX[11] being the current MSD of the multiplier. To effect this decrement, subtraction is enabled [GD53], input of RY to the AU B input is disabled [MA15.2] in favor of a 1 presented to the B input for the first bit of the digit-phase [MO6]. The carry output of the AU influences this through [UC3.6] but will be 0, this is inverted to 1 by [UC4.11].

The above addition and decrement are performed repeatedly by implicit looping of State 6. The looping terminates when RX[11] is decremented to zero. This is detected by observing RX, N8, N4 and N2 [GD159,GD120] at ØEON. At first thought, one might think to observe N8,N4,N2 and N1, but at ØEON when a state transition is being determined, the final bit shift of RX has yet to take place, so the Machine looks one bit ahead in the bit-stream. When zero is observed, the loop exits to State 7.

• State 7 *Check for loop end*

After 12 iterations of the primary loop the digits of the multiplier have necessarily been exhausted and it is time to exit the loop. The 12 traverses through State 5 in the course of these iterations rotated the PLT mark up 12 times. The mark will then be emerging from the PLT shift register, and asserting PLT, during ØD11. As the ØD11 shift is suppressed, PLT remains asserted during ØD12, where it is detected at the end of State 7, and causes the primary loop to exit to State 9 [MO7].

• State 9 *Adjust for DP*

The product is now complete in regards to its numeric digits, however if there were decimal places in the multiplicand, the product must be divided by 10 in accordance with the number of those decimal places, e.g. 10•2=20, while 10*0.2=2.

RX is placed on the 12-digit data loop [MA13.2]. If DPC≠N, the ØD12 shift is not suppressed [UC2.3], giving a 13-digit cycle: RX is shifted down one digit. The DPC is decremented. If DPC=N, the ØD12 shift for RX is suppressed giving a 12-digit cycle: RX is unaltered.

The RX shift down and DPC decrement are repeated by looping of State 9 until DPC=N, at which point the multiplication is complete and the Machine proceeds to State 0 [GD80].

Division

Division is performed using an algorithm analogous to long-division. The divisor (RY) will be shifted up to RY[9], then subtractions from the dividend in RX performed until an overdraft occurs. For each subtraction prior to the overdraft, RX[11] is incremented, resolving a digit of the quotient. The overdraft is then corrected via a restoring addition, and RX is rotated left one digit. This shifts the dividend up in preparation for more subtractions, shifts the building quotient up, and rotates the newly-resolved digit into the quotient.

• Loops — There are three loops in the algorithm: a loop to shift the divisor up (State 4), a sub-loop within which the subtractions and increments occur (State 6), and a primary loop within which the sub-loops and shifting occur (States 6,7,5).

• PLT use — The number of iterations of the primary loop is a function of the number of digits and number of decimal places in the divisor.

nd(RY) = number of digits entered into RY

dp(RY) = number of decimal places entered for RY

primary loop iterations = 11 – nd(RY) + dp(RY)

The point of involving the number of divisor decimal places in the iteration count is to multiply the quotient, e.g. 10÷2=5, but 10÷0.02=500.

The Primary Loop Tracker is used to track that portion of the iterations attributable to the number of digits in the divisor. Here, the PLT may be seen as a FIFO stack of marks, marks being pushed down into the PLT as the divisor is shifted up, then pulled up during iterations of the primary loop.

For each iteration of State 4 as the divisor is being shifted up, a 0-mark is pushed down into the PLT. The number of marks then accords to the number of those shifts. In the primary loop of States 6-7-5, the DPC is first decremented on each iteration, until the DPC reaches 0 (and transformed to N). On subsequent iterations the PLT marks are pulled up, until exhausted, causing the primary loop to terminate.

• Preconditions — Prior to the division process proper the following preconditions exist:

- the dividend has been entered into RX and aligned to the DP setpoint,
- the divisor has been entered into RY,
- the DPC has been set to the number of decimal places of the divisor (RY),
- OPMD and OPSD have been registered by the press of the division key, these assert OPD.

• Quotient — Individual digits of the quotient will be built by increment in RX[11] and then be rotated into the lower digits of RX to form the quotient.

• State 1 — *Inject a first mark into the PLT*

The PLT having been previously cleared to all-1s, a single-bit 0-mark is injected into the PLT shift register during ØD12 [GD139].

• State 4 — *Shift up divisor and push marks into the PLT.*

RY is placed on a 13D/12Ø loop [D98,UC2.9], shifting RY up one digit.

For the PLT, shift on ØD11 is not suppressed [GMO5.2.200K], giving a 13-digit cycle. Marks thus rotate down one digit-phase. A 0-mark is injected into the PLT shift register at ØD12 [GD149]. Marks do not overlap due to the extra shift at ØD11, so a new bit is marked.

State 4 loops until the divisor is shifted up enough that its MSD enters RX[9]: A 1-bit passing through the bitstream at N1 during ØD10 [GD83] sets the DFLG FF low (DFLG asserted) [UB2.12]. The FF holds the indication for detection at the end of the state cycle, thus a non-zero digit causes the loop to exit and take the Machine to State 6 [GD91]. Detection is during ØD10 rather than ØD9 because N1 is 4 bits further along in the bitstream than the output from RY.

The DFLG FF is reset high at the beginning of every number cycle [D81], relying on drain-triggerring of the FF.

This loop-until-non-zero is the source of the infinite loop when attempting to divide by 0.

• State 6 *Subtract and increment.*

Subtraction is enabled during ØD0::11 [GD48,GD54] to subtract the divisor (RY) from the reducing dividend (RX). If the subtraction results in an overdraft, a borrow propagates out of the AU carry output at the beginning of ØD12 [UE4.3], thru UC3.6, to assert R<0.

If there is no overdraft (R<0=L), subtraction is disabled [D46] and a 1 is injected into the AU B input [MO6,UE4.12], to add 1 to the building quotient in RX[11], the 1 coming from R<0=L being inverted [UC4.11].

If there was an overdraft (R<0=H), the internal borrow of the AU must be suppressed [GD88,UE4.6] to keep RX[11] from being altered, and the sub-loop exits to State 7 [MO9].

• State 7 *Correct overdraft.*

RX and RY are both fed to the AU [MA15.4], with RX on a 13D/13Ø loop [D82] and RY on a 12D/12Ø loop. RY is thus added back into RX to undo the overdraft, with RY unaltered.

If the DPC≠N, it is decremented and transformed to N if 0. See DPC commentary for details.

The Machine proceeds to State 5 [D110].

• State 5 *Shift dividend and quotient up, rotate new digit into quotient. Assess primary loop iterations.*

RX is looped through the AU on a 13D/12Ø loop, the shift on ØD12 being suppressed [UC2.4], RX is thus rotated left one digit. In one action, this shifts the dividend up, shifts the quotient up, and rotates the newly-resolved digit into the lower end of the quotient.

For the PLT, shift is suppressed during ØD0 [GD154] in addition to the usual ØD11 [D153], giving an 11-digit cycle: the PLT rotates up a digit-phase, which pulls up the 0-marks.

However, in counting primary loop iterations, the DPC is to be dealt with before the

PLT. To this end, if DPC≠N, the ØD11 shift is not suppressed [MO5], giving the PLT a net static 12-digit cycle.

Thus, in iterations through the primary loop, the DPC is first decremented to 0 and transformed to N during executions of State 7, with PLT mark pulls being suppressed in State 5. When DPC=N, suppression of the PLT ØD11 shift is reinstated, and the PLT marks proceed to be pulled.

The Machine proceeds to State 6 [D90] while PLT marks are present, but eventually the mark pulls cause PLT to not assert at ØEON, and the primary loop exits to State 9 [MO8], with State 6 being suppressed via [MA14.3,GMA11.3].

• State 9 *Clear remainder.*

The final overdraft and correction may have left a remainder from the dividend in the digits of RX. These need to be cleared to leave only the quotient.

Remarkably, the state of the PLT provides an indication of which digits need to be cleared. 0-marks pulled up from the PLT did not disappear off the 'top of the stack', rather, they were rotated around to lower digit-phases in the PLT. The PLT bits which remain unmarked (1) are the digits that need to be cleared.

RX is placed on a 12D/12Ø loop [MA13.2,UC2.3]. (Note: DPC=N by this time). Recirculation of the RX bitstream is shut off during digit-phases where there is no 0-mark in the PLT [GD109], zeroing those digits of RX. This is accomplished in a single pass throught State 9, after which the division is complete and the Machine proceeds to State 0 [GD80].

———————

Revision Log:

• 2020 May: Initial creation.

• 2020 Oct: Description of PLT operation improved, example of AU internal operation added.

• 2022 Feb: Divisor=>dividend correction in Division Procedure - State 9.

• 2022 Mar: Error flag condition previously not understood resolved. Mention of TEAL and abc-800 added.

—— End ——