bcc	title	prefix/class-number.revision
	PHASE 1.5 CPU ADDER/CYCLER CARD	1.5ACC/W-42

checked	authors	approval date	revision date
Ulrich Spormann	Max Marutani	4-3-70	
checked: C. Thacker	Max Marutani	classification: Working Paper	
approved: Jack T. Hastie		distribution: Company Private	pages: 8

ABSTRACT and CONTENTS

The adder/cycler card for the Phase 1.5 CPU is functionally similar to the standard microprocessor adder/cycler (ref. MPACC/M-2) with the exception that a left cycle of zero (LCYØ) which produces BL V BR on the X-bus is available in the Phase 1.5 version.

This document describes the functional characteristics and hardware implementation of the Phase 1.5 CPU adder/cycler.

FUNCTIONAL CHARACTERISTICS:

1. The Phase 1.5 CPU adder/cycler is bit-sliced into 3 boards of 8 bits each in a manner similar to the standard adder/cycler.

2. Instead of obtaining the LCY control signals from the control logic and special function cards, the special functions field (MSØ-MS5, high and low true) and the least significant five bits of the Z-bus, Z(19)-Z(23), are received and decoded in the following way:

MS=Ø	LCY Ø	(all left cycles by Ø produce BL+BR, not just BL).
MS=1	LCY 1	
MS=2	LCY 2	
MS=3	LCY 3	
MS=4	LCY 4	
MS=5	LCY 8	
MS=6	LCY 12	
MS=7	LCY 16	
MS=1ØB	LCY 2Ø	
MS=11B	LCL Z	(as in the standard adder/cycler except left cycles by Ø produce BL+BR)
MS=12B	LCH Z	

3. TAX gates the adder output onto a double-buffered X-bus and disables the cycler output. With the advent of the valid LCYØ, the TAX A and TAX B functions which enter the standard adder/cycler to disable the cycler are no longer received or generated in the Phase 1.5 version.

4. TCX transfers the contents of CØ through C23 onto the X-bus as in the standard adder/cycler.

5. The six adder carry signals exiting the standard adder/cycler are replaced by a single high true carry indicator, Cθ, which serves to set the carry latch for S.F.57B on the CPU 1.5 special functions #2 board.

6. The Phase 1.5 cycler has been designed to allow a left cycle and load of any register (including scratchpad) in one cycle. The Phase 1.5 adder has been designed to allow an add and branch in two cycles. VCY should be set on all adds.

HARDWARE DESCRIPTION

ADDER: The Phase 1.5 CPU adder was designed to allow an add and branch on X-bus result in two cycles. The amount of time allowed for the adder to produce a high true sum on X from the low true BOOL inputs is found as follows: (refer to figure 1)

1.	td = Ø	leading edge of high true clock on I register board
2.	td = 22/14	BL', BL, BR', BR exit I register
3.	td = 25/17	BL', BL, BR', BR enter MQ
4.	td = 57/38	BL()', BR()' exit MQ
5.	td = 60/41	BL()', BR()' enter adder/cycler
6.	td = 60/41 + t_{ad}	Sum exits X-bus at adder/cycler
7.	td = 63/44 + t_{ad}	Sum enters branch condition #1
8.	td = 87/64 + t_{ad}	BRANCH' exits branch condition #1
9.	td = 90/67 + t_{ad}	BRANCH' enters control logic
10.	td = 106/83 + t_{ad}	TBO' exits control logic
11.	td = 109/86 + t_{ad}	TBO' enters OOS
12.	td = 135/1Ø8 + t_{ad}	Branch result available for load O Reg signal on OOS

t_{ad} = 200 NS - 135/108 NS = <u>65/92</u> NS

In order to approach the worst case add time of 65 NS as well as containing the can count on each 8-bit board to below eighty, the signetics 8260 Arithmetic Logic Element was employed. This chip contains a 4-bit adder with carry look-ahead for each bit and logic for anticipated carry across adder blocks. As shown in Figure 2, two chips are used for each 4 bits to be added. One chip assumes a zero carry is entering the block while the second chip assumes a one carry. The "zero-carry sum" and "one-carry sum" are then generated by each respective chip and when the actual carry becomes

available, it does not enter the adder but instead gates the appropriate sum out to the X-bus.

Figure 3 illustrates the implementation of carry acceleration between boards. This gating was designed to handle the carry anticipation for the two most significant adder blocks, P and Q, but is also used for the blocks of lesser significance (L-Θ).

Figures 2 and 3 also present maximum and typical gate delays and illustrate how the carry paths are the limiting factor in adder speed. The adder delay (BL()' and BR()' input to X-bus output) is composed of:

1. t_{ad} = Ø — BL ()', BR ()' enter adder/cycler
2. t_{ad} = 25/12 — carry signals exit each chip
3. t_{ad} = 33/20 — carry signals exit adder/cycler board
4. t_{ad} = 36/23 — carry signals enter next adder/cycler board
5. t_{ad} = 51/34 — enable zero carry sum signal formed
6. t_{ad} = 57/40 — enable one carry sum signal formed
7. t_{ad} = 67/46 — conditional sum is gated with TAX and enable
8. t_{ad} = 75/54 — sum is gated onto X-bus

The combining of the worst case allowable add time and actual worst case add time indicates that about 10 NS more speed is required. It is highly improbable that the two worst cases

will occur simultaneously; however, to be cautions, all circuits in the bit Ø line will be selected to ensure that worst case figures are not approached.

CYCLES: The Phase 1.5 CPU cycler was designed to allow one period register to register or register to scratch pad cycling. Figure 4 illustrates the delays involved in register to register cycling; register to scratch pad delays are comparable. The delays are:

1. td = Ø MQZ clock leading edge
2. td = 30/25 register output Z ( ) valid
3. td = 40/35 register output exits MQZ
4. td = 43/38 Z ( ) enters adder/cycler
5. td = 95/79 X-bus valid from adder/cycler
6. td = 98/82 X-bus valid at MQZ
7. td = 109/89 register input ready for clock

The cycler path should ideally be ready by t = 100 NS and is seen to be 9 NS slow using worst case numbers. This difference may be at least partially recovered by the selection of fast gates for those paths found to be slow. Further, it is possible for the data to arrive a few nanoseconds after the register clock begins to rise, although this is not a desirable practice.

FIGURE 1

PATHS DETERMINING ALLOWED ADDER TIME

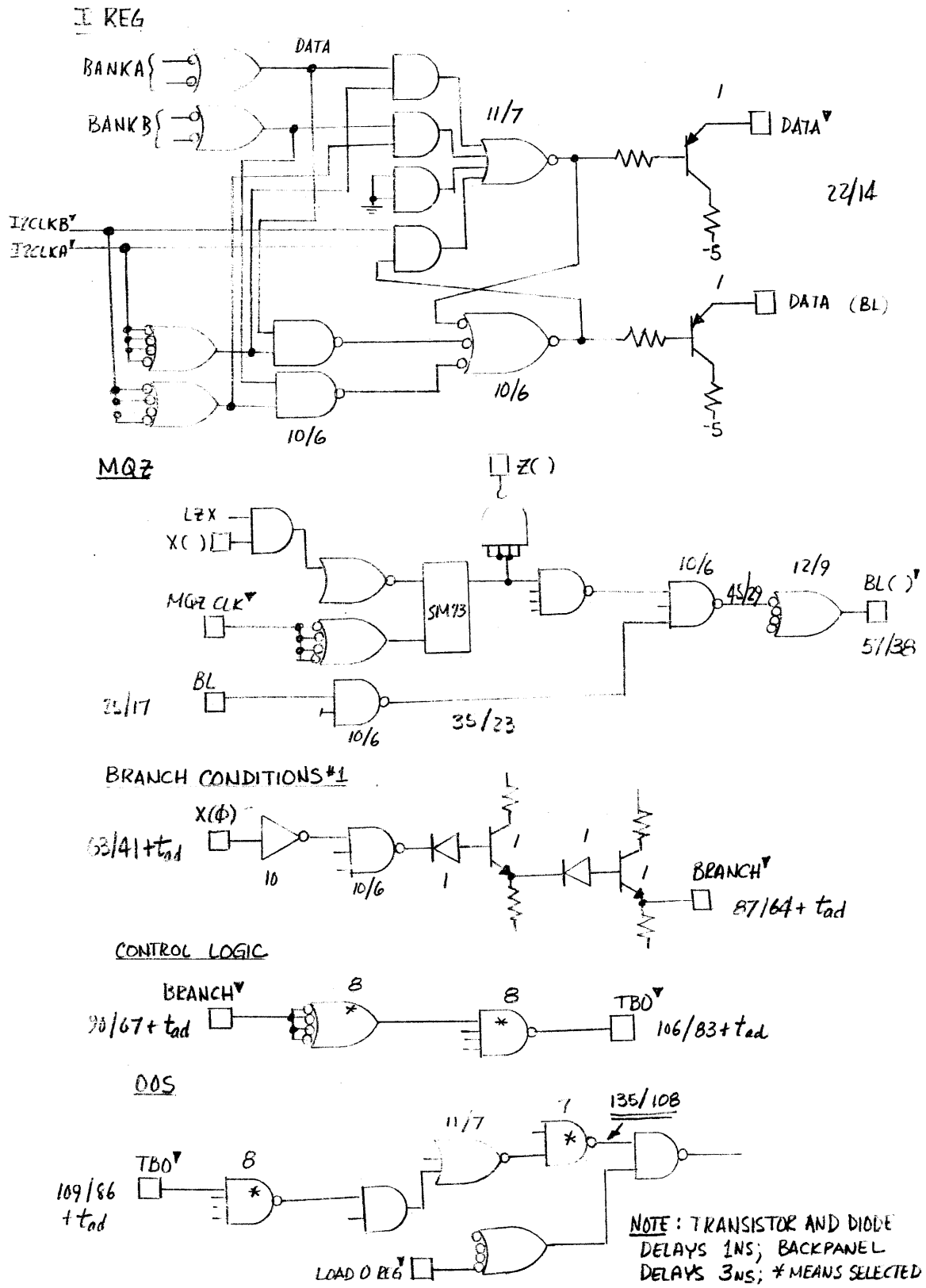


FIGURE 2

ADDER BLOCK DISTRIBUTION & ADDER DELAYS

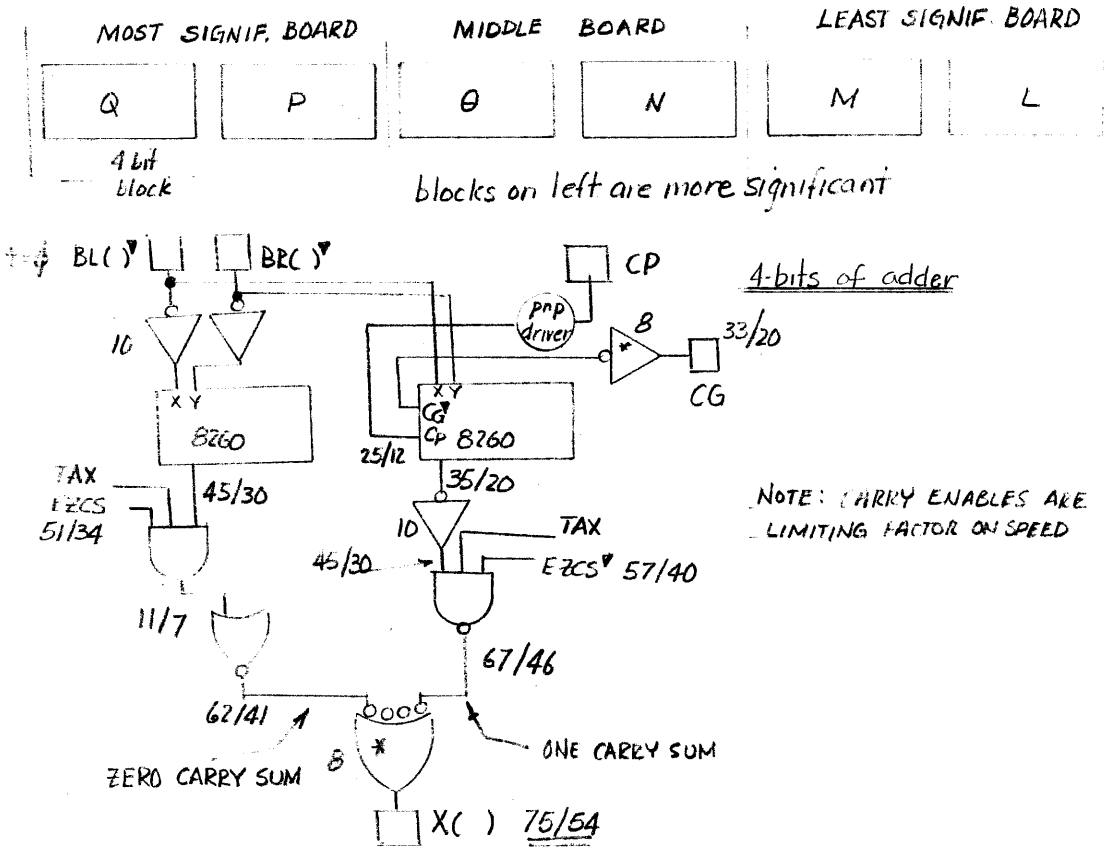
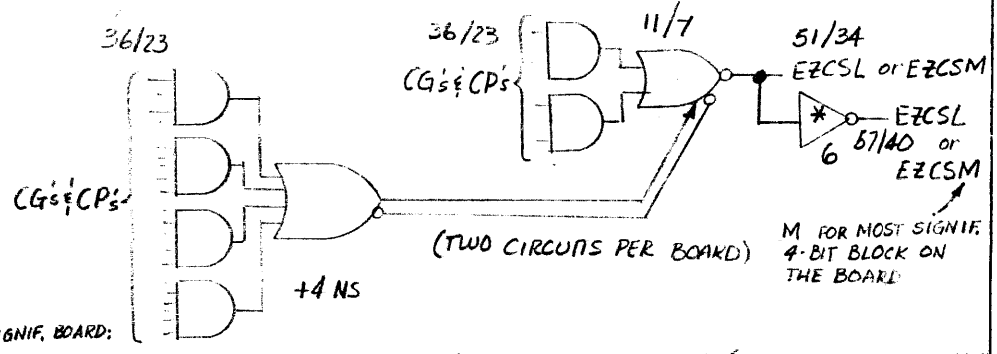


FIGURE 3

CARRY ACCELERATION ACROSS BOARDS



E.G. MOST SIGNIF. BOARD:

$EZCSL' = CGO \vee (CGN \wedge CPO) \vee (CGM \wedge CPN \wedge CPO) \vee (CGL \wedge CPM \wedge CPN \wedge CPO) \vee (LOC \wedge CPL \wedge CPM \wedge CPN \wedge CPO)$

$EZCSM' = CGP \vee (CGO \wedge CPP) \vee (CGN \wedge CPO \wedge CPP) \vee (CGM \wedge CPN \wedge CPO \wedge CPP) \vee (CGL \wedge CPM \wedge CPN \wedge CPO \wedge CPP) \vee (LOC \wedge CPL \wedge CPM \wedge CPN \wedge CPO \wedge CPP)$

FIGURE 4
CYCLER DELAYS