bcc	title	prefix/class-number.revision
	Microprocessor Adder/Cycler Card	MPACC/M-2

checked	authors	approval date	revision date
Wm Dodge	Charles P. Thacker	5-8-69	
checked		classification	
		Manual	
approved		distribution	pages
Lee Hyatt		Company Private	6

ABSTRACT and CONTENTS

This document is a description of the Adder/Cycler Card (AC-00005) used in the BCC microprocessor.

Adder/Cycler Card

The adder/cycler contains an 8-bit slice of the adder and the cycle box. The adder portion of the card is arranged logically as two four-bit sections with carry acceleration over each section as described below. A consideration in the design of the adder is that, in addtion to the time required to do the add, the delay around the loop from the M, Q, and Z registers, through the adder, onto the X bus, and back into M, Q and is 5 levels of logic plus two backpanel delays*. This amounts to (5x6) + (2x5) = 40 nsec. Since data must be stable during the clock (K2) which loads M, Q, or Z (20 nsec wide), only 80 nsec is available within a cycle for operations on registers. The adder would have had to do a full 24-bit add in 40 nsec to allow a one-cycle add. The time allowed to do a full add was accordingly increased to two cycles by requiring VCY to be set during adds, which allows 200-20-40 = 140 ns. for the full 24-bit add to occur. The carry acceleration was designed to yield a value only slightly less than this, to minimize hardware.

A single bit of the adder is shown in figure 2. The input carry, Ci is the AND of the two inputs. The $\overline{Ci}$ term is the AND of the other two inputs. The generation and propagation of carries will be discussed below. The sum is of the form:

*(Time required for a signal to propagate between cards - assumed to be 5 nsec.)

$$SUM = TAX \cdot [(\overline{Ci} + (BL.BR) + (\overline{BL}.\overline{BR})) \cdot (Ci + (\overline{BL}.BR) + (BL.\overline{BR}))]$$

TAX is only a gating term to put the sum onto the X bus, and may be disregarded.

Multiplying the sum term:

$$SUM = (BL.\overline{BR}.\overline{Ci}) + (\overline{BL}.BR.\overline{Ci}) + (BL.BR.Ci) + (\overline{BL}.\overline{BR}.Ci)$$

which, although there is some redundancy in the gating, is exactly the right thing.

Adder Carries

The carry portion of a single stage of the adder is shown in figure 3. It may do one of three things with respect to carries, namely:

1) generate a carry out independently of the carry in
2) it may propagate a carry in across itself
3) it may absorb an input carry, i.e., receive a Ci but propagate a $\overline{C}$.

Referring to the truth table for the adder (figure 1) the first case occurs when BL.BR=1. When this is the case, G=Ø, H=1, I=1, J=1, independent of the state of Ci and $\overline{Ci}$. Since C=I.J=1, and $\overline{C}$=G.H=Ø, the correct action results.

Case 3 occurs when $\overline{BL}.\overline{BR}$=1. When $\overline{BL}.\overline{BR}$=1, G=1, H=1, I=Ø, J=1, independent of the carry input. Again, the right thing happens.

The second case, that of the propagated carry, occurs when BL (EOR) BR=1. In this case, G=1, and I also =1. If there is an input carry, H=Ø and thus $\overline{C}$=Ø; if there is an input $\overline{carry}$, J=Ø and therefore C=Ø. Carry propagation requires one level per stage.

The adder is logically divided into six four-bit sections, with carry acceleration over each section (see figure 4). The acceleration works as follows: As has been mentioned, if a given stage either generates or absorbs a carry, the output carry from that stage is fully determined, independent of the carry in for that stage. Within a four bit group having no anticipation, the output carry from the high order bit is fully determined 4 logic levels after the BL+BR and $\overline{BL}$+$\overline{BR}$ signals for all the bits have stabilized (assuming that the low order bit generates a carry out, and the next 3 bits propagate the carry).

Also, the longest time required for a carry to arrive at a given stage is 4 logic levels from the time the carry arrives at the low order bit (assuming that the low order 3 bits propagate the carry, and the 4th bit absorbs the carry).

If all four bits of a group are in the correct state to propagate a carry in (BL+BR=1), or to propagate a $\overline{carry}$ ($\overline{BL}$+$\overline{BR}$=1), then the carry out from the four bit section can be determined in two levels rather than 4. The PK

signal indicates that a $\overline{\text{carry}}$ is input to a group, and is propagated across the group (i.e. the group will definitely not generate a carry). The $\overline{PK}$ signal is used as one of the CARRY inputs to the next group, and is also used to force the $\overline{\text{CARRY}}$ output of the high order bit of the group to 1. In a similar fashion, the PC signal indicates that a CARRY is input to the group, and is propagated across all 4 bits. $\overline{PC}$ is sent to the next as part of the $\overline{\text{CARRY}}$ input term, and also forces the CARRY output of the high order bit of the group to 1.

The anticipatory circuitry guarantees that a carry out from a group will be stabilized:

1) 4 levels after the BL+BR and $\overline{BL}+\overline{BR}$ signals for the group have stabilized, or
2) 2 levels after the carry signals from the next less significant group are stable at the input of the group,

whichever is greater.

The total time required (from the arrival of $\overline{BL}$ and $\overline{BR}$ at the card input) to obtain a stable carry out from bit Ø is:

- 2 levels to generate $\overline{BL}+\overline{BR}$ and BL+BR
- \+ 4 levels to generate the carries from the first group of 4 bits (bits 20-23)
- \+ 5x2=1Ø levels to propagate the carries across the remaining five groups.

This is a total of 16 levels.

From the time the carries have stabilized, 3 more levels are required to stabilize the sum on the X bus. The total is therefore 19 levels.

Cycler

The cycler is straightforward. The 8 AND inputs of the AND-NOR gates for each bit are connected to the BL signals for bits which are 1,2,3,4,8,12,16 and 20 bits to the right of the particular bit. The cycler therefore does left cycles only. The OI1-OI16 inputs for each card are wired on the backpanel to the 16 bits of the 24 bit word which are not otherwise available on the card. When the cycle count is taken from the special function field, the gating inputs for the cycler, LCY1B through LCY20B, are generated on the special function card. When the cycle count is decoded from the Z register, the LCY1A through LCY20A inputs are taken from the control logic card.

this as an AND of 2 terms from the preceding stage

BL	BR	C_{in}	SUM	C_{OUT}	
0	0	0	0	0	$\overline{BL}\cdot\overline{BR}$ generate an output $\overline{CARRY}$ unconditionally
0	0	1	1	0	
0	1	0	1	0	$\overline{BL}+\overline{BR}$:
0	1	1	0	1	BL (EOR) BR: propagate CARRY IN to next stage
1	0	0	1	0	
1	0	1	0	1	
1	1	0	0	1	BL+BR: BL·BR: generate an output carry unconditionally
1	1	1	1	1	

Fig. 1 Adder truth table

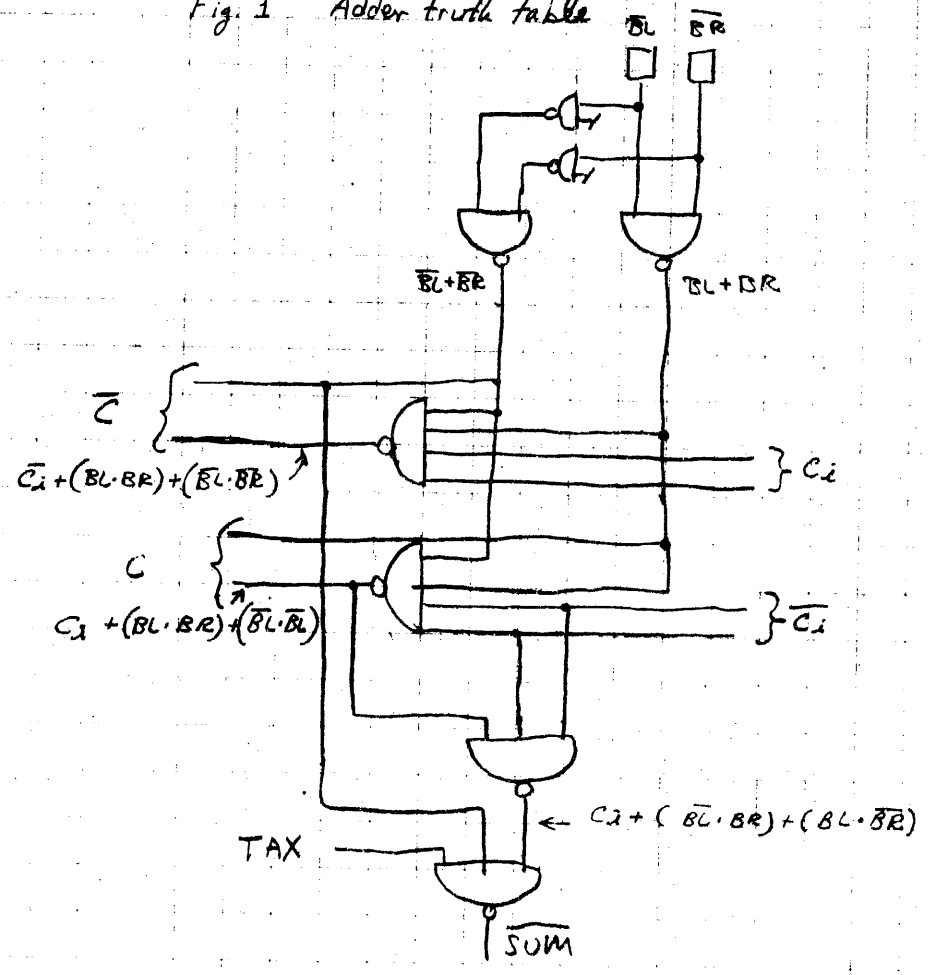

Fig 2 One bit section of the adder, with no carry anticipation

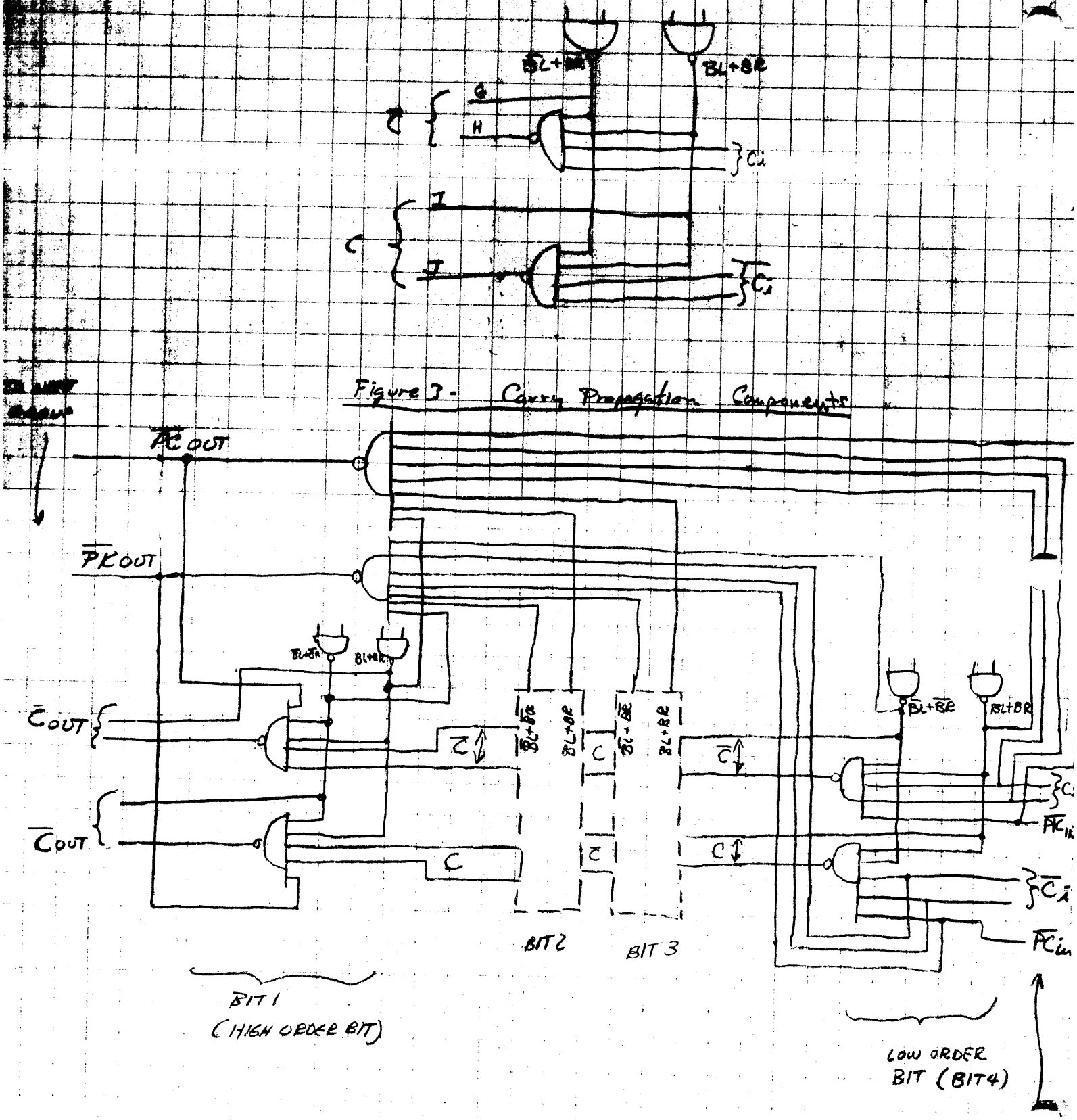

FIGURE 4 CARRY ACCELERATION OVER A 4 BIT GROUP