bcc	title	prefix/class-number.revision
	MICROPROCESSOR CONTROL LOGIC CARD	MPCLC/M-3.1

checked	authors	approval date	revision date
	Charles P. Thacker	10-2-69	
checked		classification: Manual	
approved		distribution: Company Private	pages: 10

ABSTRACT and CONTENTS

This document is an electrical and functional description of the Control Logic Card used in the BCC microprocessor. It also describes the instruction sequencing of the microprocessor.

CONTROL LOGIC CARD

The control logic card contains decoders for the scratchpad address and cycle count, the clock distribution circuitry, and the overall control logic for the microprocessor. The scratchpad address is decoded from the low order 6 bits of the Z bus, or from the SSPØ - SSP5 field, depending on whether the SPFZ special condition is 1 or Ø. The SXSn and SYSn signals go to two cards (64 words total) of scratchpad memory, and the card is selected by ADØ* or AD1*.

The cycle count may be taken from Z by executing one of two special conditions, CCFZA or CCFZB. When CCFZA is executed, Z(22) and Z(23) are interpreted as follows:

Z22 - Z23	ACTION
Ø	No cycler activity; set TAX to pass information through the adder
1	LCY 1
2	LCY 2
3	LCY 3

When CCFZB is executed, Z(19), Z(20), and Z(21) are interpreted as follows:

Z19 - Z21	ACTION
Ø	Set Tax
1	LCY 4
2	LCY 8
3	LCY 12
4	LCY 16
5	LCY 20

6 Set TAX

7 LCY 4

A microprocessor instruction may require one, two, or three hundred nsec. to execute. The control logic has three states, each state corresponding to one of the three possible intervals in an instruction. The types of instruction are as follows:

1. 'Normal' instructions. These do not branch, and do not have the VCY bit set. Execution occurs in state A, and requires only 100 nsec. Timing is shown in figure 1a.
2. 'Stretched'instructions. These do not branch, but have VCY set. They use state A and state B, and require 200 nsec. The actual execution of the instruction (i. e. loading of a register specified by the instruction) occurs at the end of state B. During state A, nothing happens. This type of instruction is usually used to allow signals to propogate to or from external devices. Timing is shown in figure 1b.
3. Normal branches. These are instructions for which the branch condition is satisfied. They use both state A and state B, and thus require 200 nsec. Register loading is done at the end of state B, but the O register (the micro-program counter) is loaded with the branch destination address at the end of state A. At the end of state B, the O register is loaded again, this time with the branch destination address +1. Simultaneously, the I register is loaded with the instruction which is the destination of the branch. Timing is shown in figure 1c.

4. 'Stretched' branches. These are branches which succeed, and which have also VCY set. They are the only type of instructions which use state C, and they require 300 nsec. for execution. This type of instruction is used only where either a) the branch condition, or b) the jump address requires a long time to generate. An example is a zero test on the adder output. Loading of any register specified in the instruction occurs at the end of state C.

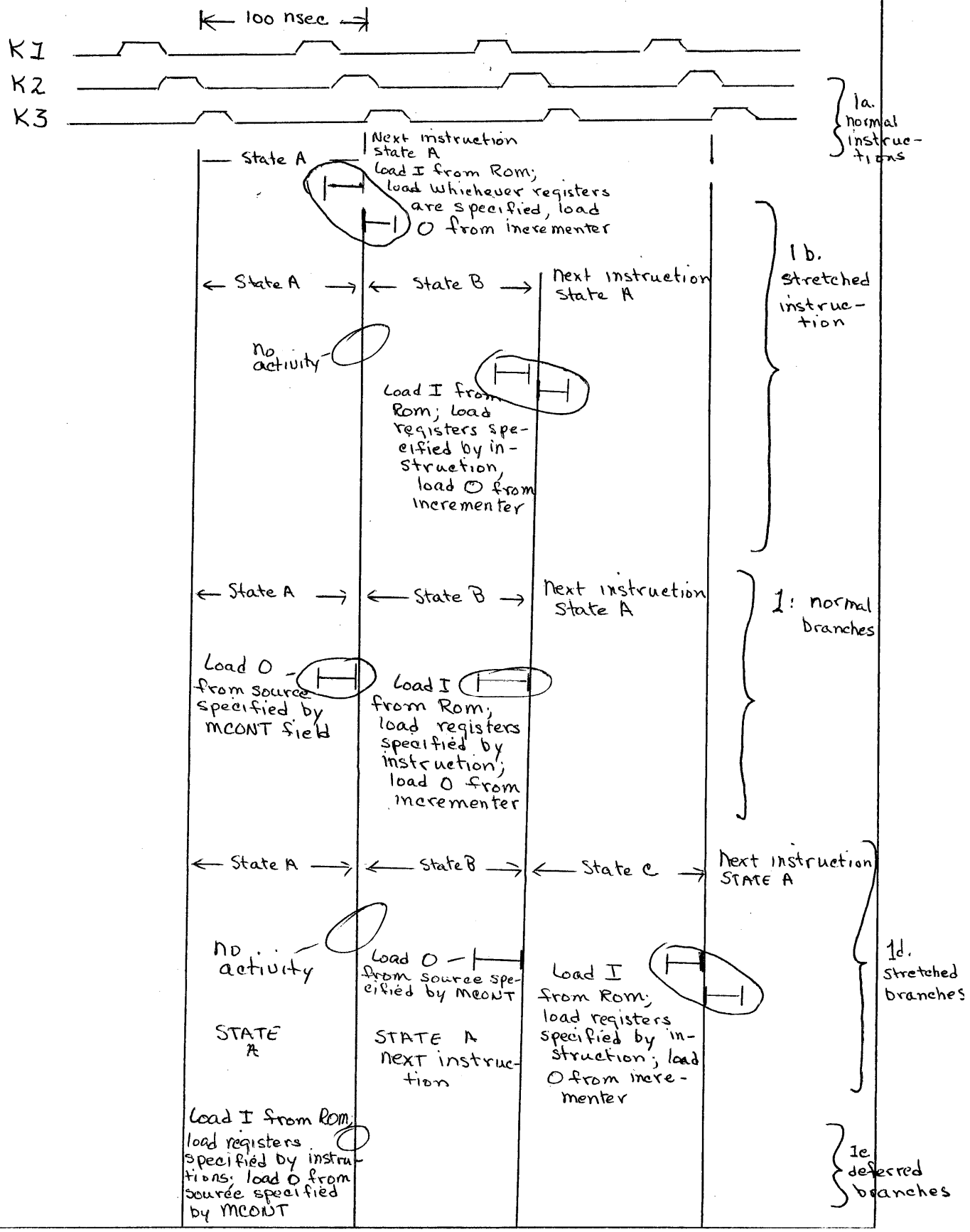
5. Calls. Subroutine calls store the O register in a separate register (OS) before loading O with the jump address. At the time OS is loaded, O will contain the address of the current instruction plus 1.

6. 'Deferred' branches. This type of instruction uses state A only (unless VCY is set, in which case state B is also entered). The only difference between a deferred branch and a normal instruction is that the O register is loaded from the B field, X, or OS at the end of state A instead of from the O register incrementer. The result of this is that the instruction after the branch is executed before branching occurs. If VCY is set, deferred branches take 200 nsec., but a full 100 nsec. is available to prepare the branch condition or the jump address. Deferred branches are done by setting the DGO bit of the micro-instruction. Timing is shown in figure 1 e.

The control logic card generates a number of system clocks from the global clock. These clocks are independently adjustable in width and delay by connecting jumpers on the card to tap on a delay line. The time of the rising edge of the true clock is set by the START tap position, the time of the fall is set by setting the STOP tap position. Each START and STOP is identified on the pc card by a single letter. The clocks are summarized below.

Name	START Tap	STOP Tap	Use
KIØ	N	M	Clocks control logic flip flops XXB and XXC
KI1	O	I	Gated to produce RØCLOCK-R6CLOCK, MCLOCK3, QCLOCK, ZCLOCK, PFL
KI3	J	H	Gated to produce I3CLOCKA and I3CLOCKB
KI4	B	L	Gated to produce LOAD OSREG
K2'	D	E	Used on special function card, parity card, and memory interface
K2X'	K	G	Used in memory interface
K3'	C	F	Used in memory interface
LOAD OREG	P	R	Generates LOAD OREG

Figure 1: Control Logic Timing



A state table of the control unit is shown in figure 2. The inputs for the state counter are VCY and DGO, which are bits in the micro-instruction, and BRANCH, which is an AND of the branch condition with the decoded branch field for that condition. The state counter is composed of flip-flops XXB and XXC. The states are:

State A = XXB' · XXC'

State B = XXB · XXC'

State C = XXB' · XXC

The transitions between states are:

SET XXB = XXB' · XXC' · (VCY + BRANCH · DGO') · SC'

CLEAR XXB = XXB

SET XXC = XXB · VCY · DGO' · BRANCH

CLEAR XXC = XXC

The state counter is clocked with KIØ.

SC is a signal generated by the memory interface. If the memory is operating and an instruction which would interfere with its operation is attempted, SC is generated. It suspends all execution until the memory stops.

All the system clocks are gated with combinations of the state and the VCY, DGO, and BRANCH signals. The RCE (register clock enable) gates the clocks for the M, Q, Z registers, the seven holding registers, and the I register. RCE is also sent to the memory interface. It signifies that the current cycle is the last cycle of the current instruction.

$$RCE = XXB' \cdot XXC' \cdot VCY' \cdot BRANCH'$$
$$+ XXB' \cdot XXC' \cdot VCY' \cdot DGO$$
$$+ XXB \cdot VCY'$$
$$+ XXB \cdot BRANCH'$$
$$+ XXB \cdot DGO$$
$$+ XXC$$
$$+ RESET$$

Additionally, SC' is ANDed with all clock signals at their respective drivers.

The clock for the O register is generated as follows:

$$LOAD\ OREG = SC' \cdot (XXB + XXC + VCY') \cdot \underbrace{R \cdot P'}$$

Clock start and stop from delay line

The only time O loading is inhibited is during the first cycle of a 'stretched' instruction. The OS register clock is generated only if the instruction is a CALL (MCONTØ = Ø , MCONT1 = 1). It is generated during state A or state B, depending on whether VCY is set.

$$LOAD\ OSREG = \{XXB' \cdot XXC' \cdot VCY' \cdot BRANCH \cdot MCONTØ' \cdot MCONT1 \cdot SC' \cdot KI4\} + \{XXB \cdot VCY \cdot BRANCH \cdot MCONTØ' \cdot MCONT1 \cdot KI4\}$$

The gating signals for loading the O register are also generated on the control logic card. The normal source of O register data is the O register incrementer. During branches, the source of O data is specified by the MCONT field. The gating signals are:

= TINCOA

TINCO = BRANCH' + (XXB · VCY') + XXC

The OR is done at the line receiver on the O register card.

TBO = BRANCH · MCONTØ' · (XXB' · XXC' · VCY' · SC' + XXB)

TXO = BRANCH · MCONTØ · MCONT1 (XXB' · XXC' · VCY' · SC' + XXB)

TOSO = BRANCH · MCONTØ · MCONT1' (XXB' · XXC' · VCY' · SC' + XXB)

	VCY	DGO	BRANCH	REGISTER CLOCKS	O CLOCK	OS CLOCK	NEXT STATE	O REGISTER DATA SOURCE	
STATE A:	Ø	Ø	Ø	1	1	1	A	INC	
	Ø	Ø	1	Ø	1	Ø/1	B	(MCONT)	← OS CLOCK =1 IF INSTRUCTION IS A CALL
	Ø	1	Ø	1	1	Ø	A	INC	
	Ø	1	1	1	1	Ø/1	A	(MCONT)	← OS CLOCK =1 IF INSTRUCTION IS A CALL
	1	Ø	Ø	Ø	Ø	Ø	B	-	
	1	Ø	1	Ø	Ø	Ø	B	-	
	1	1	Ø	Ø	Ø	Ø	B	-	
	1	1	1	Ø	Ø	Ø	B	-	
STATE B:	Ø	Ø	Ø	-	-	-	-	-	← NOT ALLOWED
	Ø	Ø	1	1	1	Ø	A	INC	
	Ø	1	Ø	-	-	-	-	-	← NOT ALLOWED
	Ø	1	1	-	-	-	-	-	← NOT ALLOWED
	1	Ø	Ø	1	1	Ø	A	INC	
	1	Ø	1	Ø	1	Ø/1	C	(MCONT)	← OS CLOCK =1 IF INSTRUCTION IS A CALL
	1	1	Ø	1	1	Ø	A	INC	
	1	1	1	1	1	Ø/1	A	(MCONT)	← OS CLOCK =1 IF INSTRUCTION IS A CALL
STATE C: (only one condition possible)	1	Ø	1	1	1	Ø	A	INC	

Other incidental signals which leave the control logic card are:

1) BRTEST = SC' · XXB' · XXC' · VCY' + XXB · VCY

   This signal = 1 during the interval in which the control logic is testing BRANCH. It is used by the branch card to reset some attention latches.

2) WS' = LSPX · XXB

   This signal is the write strobe for the scratchpad.

3) PFL' = LPF · RCE · SC' · KI2

   This signal is a clock used by the memory interface to load the priority latches for central memory requests.

4) Note that the ICLOCK signals are gated with BSELECT. This is the way the bank of ROM to be used is selected. More on this is in MEMS-1/S-11.

5) Note that RCE exits the card on pin 42.

6) Note that STATE A', XXB' and XXC' leave the card for use by possible future hardware.

Hardware funnies, oversights and errors on the control logic printed circuit card.

1) Module 4G pin 1 is connected to BRANCH. It must be connected to VCY.

2) K2', K2X', K3' leave the card from SN74HØØN's rather than SN74H40N's. Termination and loading must be checked carefully.

3) RCE and STATEA' are generated by non-SN74H40's. If used, they should be checked for termination and loading.

4) SC' is applied to some gates redundantly. If it isn't in the equations but is on the schematic, this is the reason.

5) KIØ has a 510-ohm pullup resistor to +5V. It isn't required.

6) KI2 has a 510-ohm pullup resistor to +5V. It is very probably required and should be verified.

7) XXB has 10 loads. There should be no timing, problems, but this should be checked.