bcc	title	prefix/class-number.revision
	THE PHASE 1.5 CPU PHYSICAL MAP	P1.5PM/S-34

checked	authors	approval date	revision date
M.A. Marietta	Ulrich Spannagel	5/13/70	
checked: C. Thacker	Ulrich Spannagel	classification: Specification	
approved: Ivan Sutherland		distribution: Company Private	pages: 12

ABSTRACT and CONTENTS

The Phase 1.5 CPU has a map like the Phase 1 CPU (see Specification PIPM/S-4.1), but with additional hardware which enables the CPU to scan its map for a physical page which the AMC wants to dump.

This document together with S-4.1 describes the Phase 1.5 CPU Physical Map.

The Phase 1.5 CPU has an additional map function called Scan, which is triggered by the special function SFSCAN (MS=63).

Scan is part of an instruction sequence which is executed whenever the CPU notices that the AMC is dumping a physical page.

Every time the AMC dumps a page it:

(1) stores the page number in a fixed cell of the Central Memory, and

(2) sets in the CPU a latch, called PPD (Physical Page Dump).

PPD will show up on the data returned on E1-bus from a mapping operation (MAP, MAPFETCH, MAPSR) on bit E1 (Ø) and E1(4). This is interpreted as an invalid mapping until the CPU has verified that the dumped page is not part of the CPU map at that time.

In order to test this, the page number of the dumped page has to be put into the Z-Register. Z(16) to Z(23) and the special function SFSCAN has to be executed.

Scan is implemented in hardware which addresses both map registers simultaneously and compares its contents with the page number appearing on the Z bus. SCAN also resets the PPD latch.

If this physical page is found, Scan terminates and returns on E1(5) to E1(12) its virtual page number +2, mod. 128.

Overflow bit E1 (Ø) is still false indicating that the page has been found before the completion of a full map scan.

This information can be used by programming to update the map, i.e., to set the empty flag on this particular page of the map.

If the physical page is not in the map, which should be the regular case, Scan terminates when the scan counter overflows.

The overflow bit is set and copied into E1 (Ø), indicating that the map is still valid.

In either case, the test for a valid or invalid map can be made only during the instruction <u>after</u> the one which has the special function SFSCAN. This test has to have a TE1Y and a branch on the sign bit of the Y-bus. It also needs a VCY.

SCAN - Hardware

The scanning feature involves the following hardware, most of it located on a third map control board.

1.) Latch PPD. PPD can be set from the AMC through a signal PPDA except when the CPU is executing a MAP, MAPFETCH or MAPSR or when a Scan is in progress.

PPD is reset by latch Scan or when the CPU is reset

Set PPD = PPDA·DOIT'

Reset PPD = SCAN + RESET

2.) Latch SCAN

SCAN is set at the end of the instruction which has special function SFSCAN.

SCAN enables the clocks for the scan counter and the compare register, described later. It also turns off the CPU register clocks by pulling STOPC' down.

SCAN can be reset as early as 2 cycles after it is set, namely, when the first map register location (Ø) is causing a compare (COMP1), and a READY latch indicates the validity of the comparison (see section 3).

If no comparison is achieved, SCAN is reset when the scan counter has overflowed. SCAN is also reset when the CPU is reset.

Set SCAN = SFSCAN·PPD·(MK2-10)

Reset SCAN = READY·(OVFL+COMP1+COMP2)·MK2+RESET

3.) Latch READY

The absence of READY prevents SCAN from being reset before the first comparison is done. READY sets when the first pages from both map registers have been stored in the compare registers, and resets with an I2 Clock, at the end of the instruction being executed.

READY also enables the transfer of the contents of the scan counter (QE(5) through QE(12)) and the overflow bit QE (Ø) onto the E1 bus.

Set READY = SCAN·QE(11)·(SCK1)

Reset READY = MI2CL

4.) 7-bit scan counter. This counter addresses both map registers simultaneously with bit 1 through 6. Bit 7 is used to set the overflow flip flop.

This counter is cleared during the instruction which brings special function SFSCAN:

CLINC = SFSCAN·SCAN'

It is then incremented every machine cycle until either a compare or overflow occurs. Its clock:

SCK1 = (MK1-10)·SCAN

5.) Each map register has a 8-bit compare register, which saves the physical page found, for one machine cycle. Its output is continuously compared with the dumped page on the Z-bus.

Its clock: SCK2 = (MK2-10).SCAN

6.) Overflow Flip Flop

Overflow indicates that the entire map has been scanned without the occurrence of a compare.

Set OVERFLOW = QE(5).QE(11).SCK1

Reset OVERFLOW = CLINC

Timing

It takes between .5μsec and 7.0μsec to scan both map registers.

Figure 1 shows the sequence of all important events.
Figures 2 through 6 give a detailed account of all worst case propagation delays in the critical paths.

The leading edge of the I2-Clock at the end of the instruction which brings SFSCAN is assumed to be the time reference t = 0.

SFSCAN is received at least by t = -30, such that SCAN will set at t = 12/22, where 12 is the earliest time and 22 the latest time (in nsec) to be expected.

SCAN will cause STOPC' to go low at t = 24/46, which is well within the window for turning off the CPU register clocks without pulse splitting (see figure 4).

The map register takes its address from the scan counter if SCAN is set. Since SCAN is getting set during the first cycle, the readout of the first address can become stable as late as t = 120 (see figure 2).

The readout of the second address occurs at t = 153/216 and the further readouts happen in 100 nsec intervals. There is only a 33 nsec window between a late bit from the first readout and an early bit from the second one, to catch the data from the first readout. The compare register right at the output of the map register with a clock from 120 to 140 gets loaded just within this window.

Let us now assume that the physical page we are looking for was found in address Ø of Map Register #1 and has been clocked into the compare register at t = 120.

During the following, second cycle, COMP1' will be issued by the Map Register #1. COMP1' reaches the map control card at t = 147/195 (see figure 3). At the end of the second cycle, at t = 192/204, READY gets set regardless of a compare.

READY allows COMP1' to reset SCAN with clock MK2 (210 to 230). COMP1' as source of the compare gets stored into the COMP1 Flip Flop with the last SCK2 clock (t = 220 to 240).

A third cycle is used to disable the scan counter clock SCK1 and the compare register clock SCK2. The disable signal comes at t = 228/244 (see figure 3).

The last SCK1 clock brings the scan counter to 2 at t = 200. READY transfers this counter output onto the y-bus by t = 242/286 (see figure 5). The last SCK2 clock comes at t = 220 to 240 and turns the compare signal off, but it is no longer used.

The CPU register clocks are turned on during the third cycle by STOPC' going high at t = 228/246 (see figure 3). Therefore, the CPU assumes State B during the fourth cycle.

Since VCY is set, y (Ø) can be tested during State B for a branch (see figure 6). A true branch condition will bring TBO and LOADOREG at t = 400-425.

Finally, at the end of the fifth cycle during which the CPU assumes State C, an I2-clock comes and terminates the Scan by resetting READY after t = 500.

If, instead of address Ø, address M causes a compare, then Scan goes through M+5 cycles, with the last 4 cycles being the same as in the previous case. By t = 242/282 + 100·M the y-bus will get M+2 as map register address.

If the scan counter reaches 65 decimal or 1000001 binary than OVERFLOW flip flop is set at t = 6600. OVERFLOW causes the same 4 cycles as a compare does. Thus, Scan ends in the same way as described before, with a final I2 clock at t = 7000.

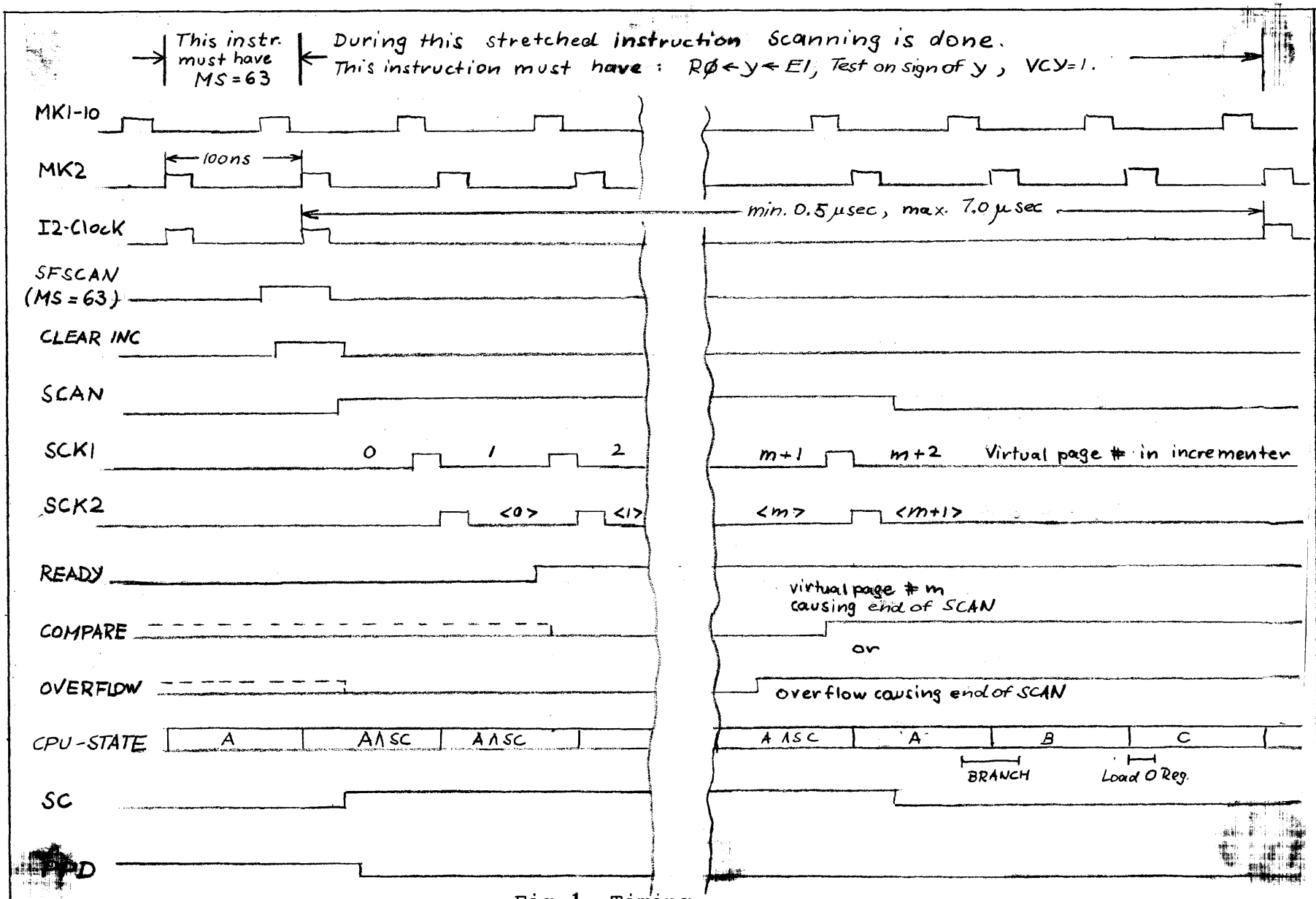
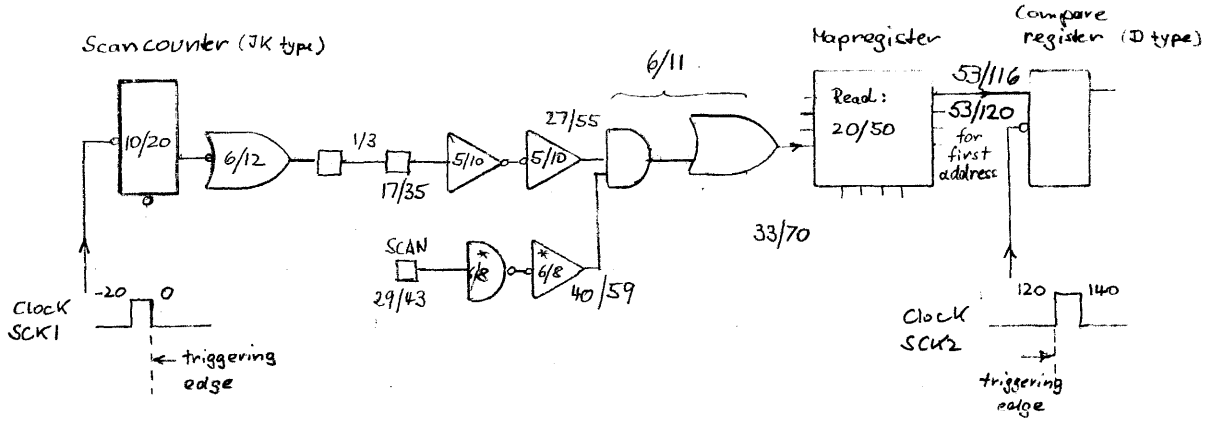

This instr. must have MS=63
During this stretched instruction scanning is done.
This instruction must have : RØ←y←E1, Test on sign of y, VCY=1.
MK1-10
MK2
100ns
I2-Clock
min. 0.5 µsec, max. 7.0 µsec
SFSCAN (MS=63)
CLEAR INC
SCAN
SCK1
0
1
2
m+1
m+2
Virtual page # in incrementer
SCK2
<0>
<1>
<m>
<m+1>
READY
virtual page # m causing end of SCAN
COMPARE
or
OVERFLOW
overflow causing end of SCAN
CPU-STATE
A
A∧SC
A∧SC
A∧SC
A
B
C
BRANCH
Load 0 Reg.
SC
D

Fig. 1 Timing

Figure 2

Scan counter to Compare register (First cycle)



Note: All times given as a/b are a = fastest prop. delay expected (in nsec)
b = worst prop. delay expected

Figure 3

Compare register to disabling clocks SCK1, SCK2 (Second and third cycle)

Assuming at time t=0 the scan counter did put out a virtual page = which is causing a compare. By the time the compare is discovered and the scan counter clock is disabled, the scan counter has been incremented by 2.

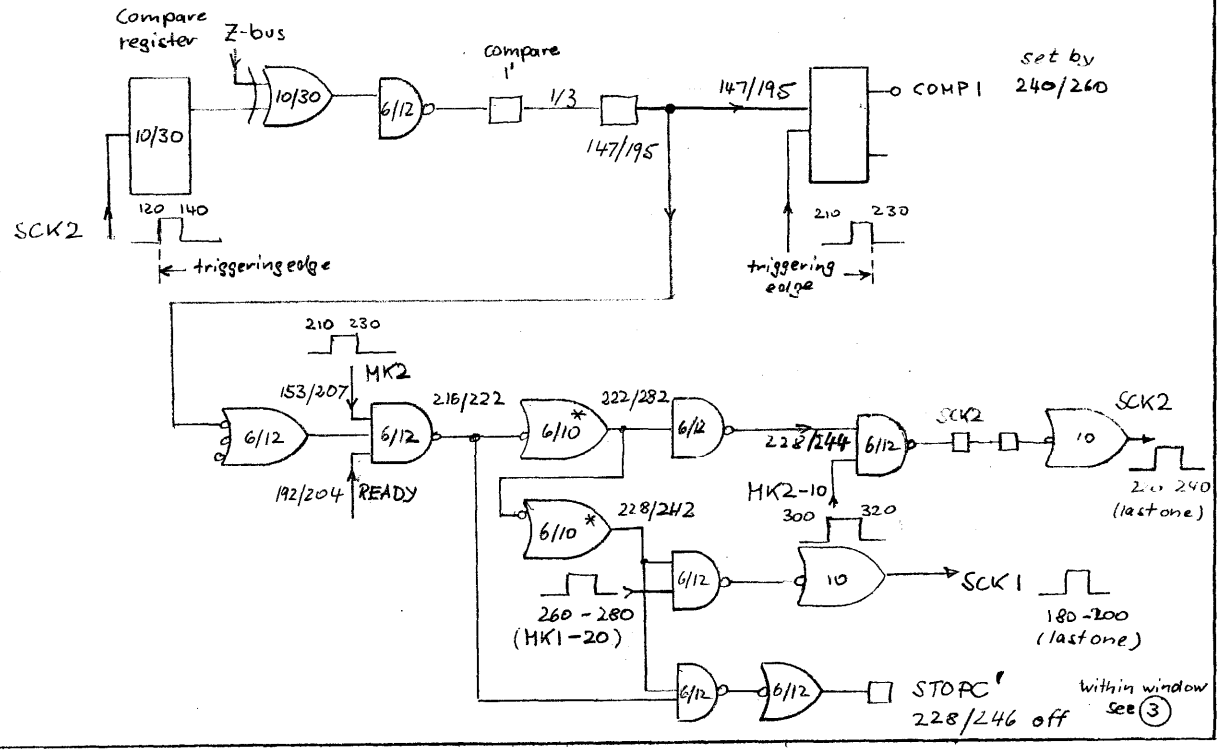


Figure 4

Stopping of CPU register clocks during scanning

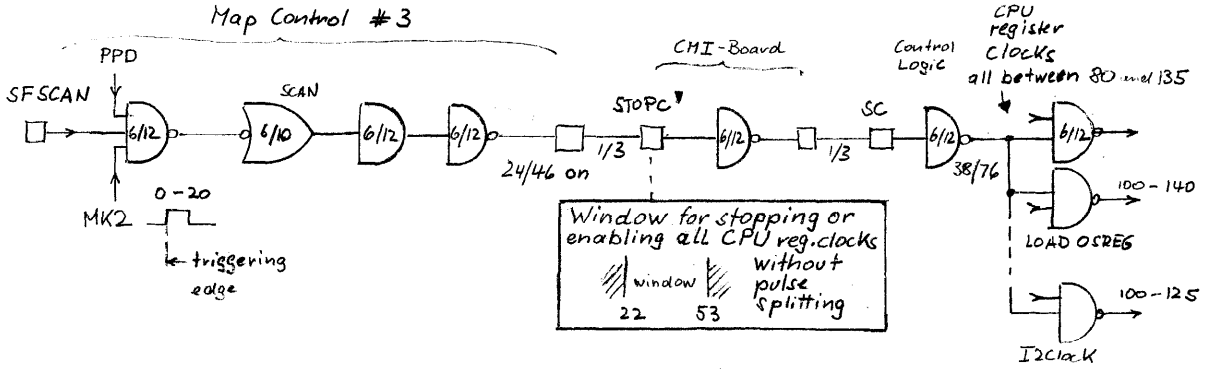


Figure 5

Scan counter to y-bus (third cycle)

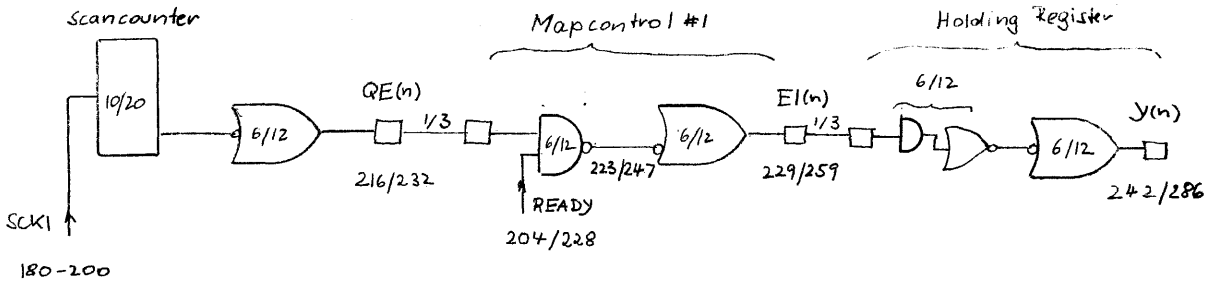
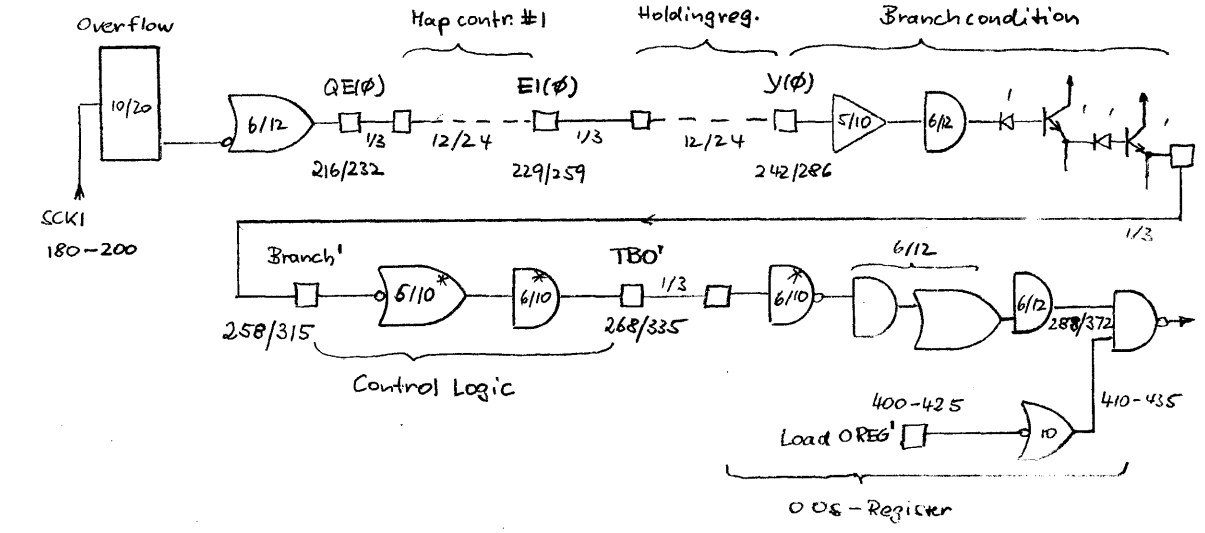


Figure 6

Branch test on QE(ϕ) (third and fourth cycle)



Appendix

A) THE CPU/1.5 bit assignment for the data returned by a MAP, MAPFETCH or MAPSR:

El bit	
Ø	MRCE + EF + PPD (ring error or empty or page dump)
1	MRCE
2	PMRO + $\overline{DB}$
3	PMRO
4	PPD (Physical page dump)
5-12	8-bit physical address from map
13-23	11-bit word address from RØ

B) The CPU/1.5 bit assignment for data returned by a SCAN:

El bit	
Ø	OVFLOW
1-4	Unused
5-12	7-bit virtual page number +2, mod. 128
13-23	Unused

c) The CPU/1.5 bit assignment for data on Z-bus when entering SCAN:

Z-bit	
0-15	Unused
16-23	8-bit physical page number of dumped page