MCS-4 Timing Detailed 4004 Timing

No No No No No No No No			A1		A2			A3	M1				M2					X1		X2	2			Х3		A1		
Part	Inst.	Cycle all al		Pclk2 a21		1 a22clk2	a31		a32clk2	m11	m12		m12clk2	m21	m22		m22clk2	x11			x21			x22clk2	x31		1 x32clk2	
March Marc			Z UTZOMI UTZ			- GEEOME		uoz uozoner		cm_rom = 0	ada = 0	i	if sc:			mezoner	if sc:	cm_rom = 0	acc_out = acc		if ope:	X22	if not io:	if io:	707	AGE AGEGIA	XOZOMZ	477 472
Mathematical Content of the conten							cm_rom = 1			cm_ram = 0	adb = 0xF		opr = data	if io:			opa = data	cm_ram = 0	cy_out = cy		alu.init() [2]		tmp = data	tmp = data				
Part	ALL [1]						cm_ram = ram bank							ram_bank														
No. Control		1/1	sc = 1																									
March Marc		1/2	sc = 1																									
The content of the	JCN	2/2										,	pm = data				pl = data											
March Marc																												
No.	FIM	2/2	sc = 0									ı	regp[opa][7:4] = data				regp[opa][3:0] = data											
No.	SRC		sc = 1																		data = regp[opa][7:4] cm_rom = 1				data = regp[opa][3:0] cm_rom = 0			
Part																					cm_ram = ram_bank				cm_ram = 0			
10	FIN																				data = regp[0][7:4]			pm = data	data = regp[0][3:0]		pl = data	\perp
March Marc	IIN												regp[opa][7.4] – data				regp[opa][3.0] – data				data = regn[ona][7:4]			nm = data	data = regn[ona][3:0]		nl = data	
March Marc	JIN																				data regptopaj(r. ij			piii data	data regptopulto.ej		pi data	
March Marc	JUN		sc = 0										pm = data				pl = data				data = opa			ph = data				+ + -
Part			sc = 1																									
March Marc	JMS		sc = 0									,	pm = data				sp				data = opa			ph = data				
Part	INC		sc = 1						1								pi – uala	+			data = reg[opa]	Н	adc = 1		data = add	+	reg[opa] = data	
March Marc			sc = 1														1			1		П	adc = 1		data = add		_	
March Marc	ISZ		sc = 0									1	pm = data				pl = data											
Married Marr									1						+ +		+	1		 	data = reg[opa]	H				+ + -		
Second Color	ADD	1/1	sc = 1						-	-							+	1		+ + -	data = regional	-						++-
No.	SUR	1/1																			uata – regiopaj		adb = ~adb		cy = co			
No. No. No.			sc = 1																		data = reg[opa]		auc – ~cy		acc = add			
Second S			sc = 1																		-						reg[opa] = data	
Martin M			sc = 1			1											sp	1		sp	data = opa			SD	_			+ + -
West Marker Mar																												
Windows Wind			sc = 1																		data = acc_out							
West 15 16 17 18 18 18 18 18 18 18	WMP	1/1	sc = 1																		data = acc_out							
West 19 19 19 19 19 19 19 1	WRR	1/1	sc = 1																		data = acc_out							
Wind	WR0	1/1																			data = acc_out							
No.																												
SMM 11																					-							\perp
SM 11	WR3	1/1																1			data = acc_out			ada = acc				acc = add
ROM 11	SBM	1/1																						adb = ~adb				cy = co
ROP 11			sc = 1																					uuc – cy				acc = add
ADM 1/1 Se-1 S S S S S S S S S			sc = 1																									acc = add
RDD 1/1	ADM	1/1	sc = 1																									acc = add cy = co
RO 11			sc = 1																					uuo oy				acc = add
RDS 1/1 0 0=1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	_		sc = 1																			П						acc = add
CLB 1/1 So-1 So -1 So -1		1/1																										acc = add
CLC 1/1 Se-1	RD3	1/1																				Ш						acc = add
Tack 11	CLB	1/1	sc = 1																									
AC	CLC	1/1	sc = 1																						cy = co			
CMC 1/1	IAC	1/1	sc = 1																									
CMA 1/1			sc = 1																				adb = ~adb					
RAL 1/1 So = 1 So			sc = 1							+					+		+			+ + -		-			acc = add			
RAR 1/1 sc=1 sc=1 ada = acc (acc, cy) = shr(acc, cy) acc = add cy = cy = co TCC 1/1 sc=1 sc=1 adc = cy acc = add cy = co cy = co DAC 1/1 sc=1 sc=1 adc = cy acc = add cy = co cy = co TCS 1/1 sc=1 sc=1 adc = cy acc = add cy = co cy = co STC 1/1 sc=1 sc=1 adc = cy acc = add cy = co cy = co DAA 1/1 sc=1 sc=1 adc = cy acc = add cy = co cy = co KBP 1/1 sc=1 sc=1 adc = cy acc = add cy = cy = co cy = co																	1			1								
TCC 1/1			sc = 1																				ada = acc		(acc, cy) = shr(acc, cy)			
DAC 1/1 sc=1 ada = acc add cy = co tos(acc_out, cy_out) cy = cy	TCC	1/1	sc = 1																				adc = cy					
TCS 1/1 Sc = 1 adc = cy acc = tcs(acc_out, cy_out) cy = 0 STC 1/1 Sc = 1 adc = cy acc = tcs(acc_out, cy_out) cy = 0 DAA 1/1 Sc = 1 ada = acc (acc, cy) = (acc,			sc = 1																	1		H	ada = acc		acc = add			
Cy = 0			sc = 1			+			 			-			+		+	+		+ + -		_				it)	1	
DAA 1/1 sc = 1 ada = acc (acc, cy) = daa(acc_out, cy_out) daa(acc_out, cy_out) KBP 1/1 sc = 1 acc = kbp(acc_out) acc										-					$\perp \perp$			1							cy = 0			$\perp \perp \perp$
DAA 1/1	STC	1/1													++					 		-				+		+
		1/1																					aua = acc		daa(acc_out, cy_out)			
DCL 1/1 sc = 1 ram_bank = dcl(acc_out)																	1	1							acc = kbp(acc_out)			
	DCL	1/1	sc = 1						<u> </u>												ram_bank = dcl(acc_out)							

MCS-4 Timing Notes

- [1] For all instructions
- [2] A complex initialization routine sets the bus with the proper value for tmp for all ope instructions.