Inst	Instructions					
No	Name	Opcode	Bits	data_in		
1	remove	000	Opcode (3bit)	(31 downto 0) id kartu		
2	sub RA, RB	001	Opcode (3bit) + mode(2 bit) + Opperand A(3 bit) + Opperand B (3 bit)	Tergantung mode		
3	mul RA, RB	010	Opcode (3bit) + mode (2 bit) + Opperand A(3 bit) + Opperand B (3 bit)	Tergantung mode		
4	mov A, B	011	Opcode (3bit) + mode(2 bit) + Opperand A(3 bit) + Opperand B (3 bit)	Tergantung mode		
5	save	100	Opcode (3bit)			
6	div RA, RB	101	Opcode (3bit) + mode (2 bit) + Opperand A(3 bit) + Opperand B (3 bit)	(31 downto 0) new time		
7	Inc RA	110	Opcode (3bit) + dont care (2 bit) + Opperand A (3 bit)			
8	Dec RA	111	Opcode(3bit) + dont care (2bit) + Opperand A (3 bit)			

	M <= Card id(32 bit) + timetstamp (32bit)	Instruction :
Get_timestamp	MOV data_out, R0	011 00 000 00
cardid_timestamp	datain(63 downto 32) <= data_out	
	datain(31 downto 0) <= card_id	
move_toreg	MOV R2, data_in	011 10 010 00
move_tomem	MOV M[R1], R2	011 11 001 01
	Inc R0	110 001 00000
	interfaceKeluar	
set_0_r2	data_in <= others 0	
	mov R2, data_in	011 10 010 00
card id to data_in	data_in(31 downto 0) <= card_id	
mov R3 data_in	mov R3, data_in	011 10 011 00
mov R4 M[R2]	mov R4, M[R2]	011 01 100 01
mov data_out R4 1	mov data_out, R4	011 00 000 10
remove timestamp	data_in(31 downto 0) <= data_out(63 downto 32)	
	data_in(63 downto 32) <= others 0	
mov R4 data_in	mov R4 data_in	011 10 100 00
sub R4 R3	sub R4, R3	001 11 100 01
mov data_out R4 2	mov data_out, R4	011 00 000 10
check data_out	if data_out = 0, next_state = mov R3 MR2	
	else, inc R2 ,next_state = mov R4 M[R2]	110 010 00000
mov R3 MR2	mov R3, M[R2]	011 01 011 01
mov data_out R3	mov data_out R3	011 00 000 01
remove cardid	data_in(63 downto 32) <= others 0	
	data_in(31 downto 0) <= data_out(31 downto 0)	
mov R3 data_in	mov R3, data_in	011 10 011 00
get_timestamp	mov data_out , R0	011 00 000 00
mov R4 data_in	data_in <= data_out	
	mov r4, data_in	011 10 100 00
sub R4 R3	sub R4, R3	001 11 100 01
div R4, data_in	data_in <= 60 * 60	
	div R4, data_in	101 10 100 00
mul R4 data_in	data_in <= 3000	
	mul R4, data_in	010 10 100 00
mov balance to reg	data_in <= balance	
	mov R5, data_in	011 10 101 00
sub R5 R4	sub R5, R4	001 11 101 10
mov data_out R5	mov data_out R5	011 00 000 10
check_balance	if data_out >= 0, balance<=data_out	
	else report uang tidak cukup	
save	1	100 00000000

Mode (mov)							
no	mode	bit	A (3 bit)	B (3 bit)			
1	data_out, reg	00	XXX	Reg B			
2	reg, mem	01	Reg A	Mem[Reg B]			
3	reg, data_in	10	Reg A	XXX			
4	mem, reg	11	Mem[Reg A]	Reg B			

Mode (aritmatika)						
no	mode	bit	A (3 bit)	B (3 bit)		
1	data_out, reg	00	XXX	Reg B		
2	reg, mem	01	Reg A	Mem[Reg B]		
3	reg, data_in	10	Reg A	XXX		
4	reg, reg	11	Reg A	Reg B		

	Registers masing-masing 64 bit						
Reg Number		bit	Reserved for				
	0	000	Timestamp				
	1	001	Used parked spot				
	2	010	-				
	3	011	-				
	4	100	-				
	5	101	-				
	6	110	-				
	7	111	-				

0	000	Timestamp		
1	001	Timestamp Used parked spot		
2	010	-		
2 3 4 5 6 7	011	-		
4	100	-		
5	101	-		
6	110	-		
7	111			