Op name	Type	Action

MEM_NOP	Memory	NOP
MEM_FETCH	Memory	Fetch and decode next instruction. Store it in INST
MEM_READ_PC_NODEST	Memory	Read data at pc and throw it away.
MEM_READ_PC_MDR	Memory	Read mem[pc] into MDR.
MEM_READ_PC_PCH	Memory	Read mem[pc] into PCH
MEM_READ_PC_ZP_ADDR	Memory	Read mem[pc] to ADDRL and zero out ADDRH
MEM_READ_PC_ADDRL	Memory	Read mem[pc] into ADDRL
MEM_READ_PC_ADDRH	Memory	Read mem[pc] into ADDRH
MEM_READ_PC_ZP_PTR	Memory	Read mem[pc] into PTR
MEM_READ_PC_PTRL	Memory	Read mem[pc] into PTRL
MEM_READ_PC_PTRH	Memory	Read mem[pc] into PTRH
MEM_READ_ADDR_MDR	Memory	Read mem[addr] into MDR
MEM_READ_PTR_MDR	Memory	Read mem[PTR] into MDR
MEM_READ_PTR_ADDRL	Memory	Read mem[PTR] into ADDRL
MEM_READ_PTR1_ADDRH	Memory	Read mem[PTR+1] into ADDRH. Incrementing pointer does not handle carry out.
MEM_READ_PTR1_PCH	Memory	Read mem[PTR+1] into PCH. incrementing ptr does not handle page crosses
MEM_WRITE_MDR_ADDR	Memory	Write MDR to mem[addr]
MEM_WRITE_A_ADDR	Memory	Write A to mem[addr]
MEM_WRITE_X_ADDR	Memory	Write X to mem[addr]
MEM_WRITE_Y_ADDR	Memory	Write Y to mem[addr]
MEM_PUSH_PCL	Memory	Store PCL at mem[S]
MEM_PUSH_PCH	Memory	Store PCH at mem[S]
MEM_PUSH_A	Memory	Store A at mem[S]
MEM_PUSH_P	Memory	Store SR at mem[S]
MEM_PUSH_P_B	Memory	Store SR at mem[S] with bit 4 set.
MEM_PULL_PCL	Memory	Read mem[S] into PCL
MEM_PULL_PCH	Memory	Read mem[S] into PCH
MEM_PULL_A	Memory	Read mem[S] into A
MEM_PULL_P	Memory	Read mem[S] into SR
MEM_NMI_PCL	Memory	Fetch PCL from NMI Low.

MEM_NMI_PCH	Memory	Fetch PCH from NMI High.					
MEM_RESET_PCL	Memory	Fetch PCL from RESET Low.					
MEM_RESET_PCH	Memory	Fetch PCH from RESET High.					
MEM_IRQ_PCL	Memory	Fetch PCL from IRQ Low.					
MEM_IRQ_PCH	Memory	Fetch PCH from IRQ High.					

DAT_NOP	Data	NOP					
DAT_INC_S	Data	Increment stack pointer					
DAT_INC_X	Data	Increment X					
DAT_INC_Y	Data	Increment Y					
DAT_INC_MDR	Data	Increment MDR					
DAT_DEC_S	Data	Decrement stack pointer					
DAT_DEC_X	Data	Decrement X					
DAT_DEC_Y	Data	Decrement Y					
DAT_DEC_MDR	Data	Decrement MDR					
DAT_MOV_A_X	Data	Copy A into X					
DAT_MOV_A_Y	Data	Copy A into Y					
DAT_MOV_S_X	Data	Copy S into X					
DAT_MOV_X_A	Data	Copy X into A					
DAT_MOV_X_S	Data	Copy X into S					
DAT_MOV_Y_A	Data	Copy Y into A					
DAT_MOV_MDR_PCL	Data	Copy MDR to PCL					
DAT_MOV_MDR_A	Data	Copy MDR into A					
DAT_MOV_MDR_X	Data	Copy MDR into X					
DAT_MOV_MDR_Y	Data	Copy MDR into Y					
DAT_CLC	Data	Clear carry flag					
DAT_CLD	Data	Clear decimal flag					
DAT_CLI	Data	Clear IRQ flag					
DAT_CLV	Data	Clear overflow flag					
DAT_SEC	Data	Set carry flag					
DAT_SED	Data	Set decimal flag					

DAT_SEI	Data	Set IRQ flag
DAT_CMP_MDR_A	Data	Compare MDR to A
DAT_CMP_MDR_X	Data	Compare MDR to X
DAT_CMP_MDR_Y	Data	Compare MDR to Y
DAT_ASL_MDR	Data	Shift left MDR
DAT_ASL_A	Data	Shift left A
DAT_LSR_MDR	Data	Shift right MDR
DAT_LSR_A	Data	Shift right A
DAT_ROL_MDR	Data	Rotate left MDR
DAT_ROL_A	Data	Rotate left A
DAT_ROR_MDR	Data	Rotate right MDR
DAT_ROR_A	Data	Rotate right A
DAT_EOR_MDR_A	Data	Apply operation to A
DAT_AND_MDR_A	Data	Apply operation to A
DAT_ORA_MDR_A	Data	Apply operation to A
DAT_ADC_MDR_A	Data	Add MDR to A
DAT_SBC_MDR_A	Data	Subtract MDR from A
DAT_BIT_MDR_A	Data	Set zero flag based on MDR & A, set bits 7 and 6 of SR to 7 and 6 of MDR.
DAT_ADD_ADDRL_X	Data	Add X to ADDRL
DAT_ADD_ADDRL_Y	Data	Add Y to ADDRL
DAT_ADD_PTRL_X	Data	Add X to PTRL
DAT_FIXA_ADDRH	Data	Add carry-out to ADDRH, add cycle if carry-out != 0.
DAT_FIX_ADDRH	Data	Add carry-out to ADDRH, never adds a cycle.
DAT_BRANCH	Data	Check if the branch was taken. If no, inc PC. If yes, add MDR to PCL and add taken cycle
----------		
N/A	BOOL	Increment PCL and add carry to PCH