|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **2.1 트랜지스터**   |  |  |  | | --- | --- | --- | | **이미지** | **해설** | | |  | 트랜지스터 | | |  | 베이스 N 또는 P 핀에  전류가 흐르게 되면  양쪽에있는 n n 또는 p p를 연결시킨다 | | |  | IC( Integrated Circuit ) 통합 회로  집접회로  트랜지스터 저항 콘덴서 다이오드 등을  하나의 칩에 설계하여 만듬.  더이상 트랜지스터를 이용하여 직접 게이트를  만들 필요가 없어짐 | | |  | IC칩의 종류에 따라  And연산게이트 IC칩  Or연산 게이트 IC칩  등등 각목적에 맞는 IC칩의 등장 | | |  | ALU( Arithmetic and Logical Unit )  산술 논리 연산 장치  각각의 역활을 하는 IC칩들을 ALU안에 전부 구현.  레지스터에서 입력을 받아모든 IC계산을 하고  opcode중 멀티플렉서 를 이용해 원하는 연산을 하고 레지스터에 연산된 값을 반환 | | |  | Controller Unit ( 제어장치 )  시스템 버스의 신호 제어 램의 oper를 읽어와라 등등 이런 신호를 Controller Unit이 전송하여  IR 명령레지스터가  명령어를 해석(바이너리코드로) 하고  명령 레지스터 에 저장한다  ( 램의 명령어를 해석 하고 읽어온다명령어를 실행)  ALU의 입력 오퍼랜드에 입력값을 전달하는 역활도함  IR( Instruction Register )  명령 레지스터  PC에서 실행한 오퍼와 오퍼랜드를 바이너리 코드로  번역하기 위해 잠시 보관 해두는 저장소 이고  번역이 된 코드는 실행을 한다.  PC( Program Counter )  프로그램 카운터  다음번 실행할 오퍼의 주소값 을 저장 | | |  | |  | CPU의 폰노이만 구조의 명령사이클  CPU가 명령어 기계어 하나를 실행하는 과정  [ Memory ] - Fetch( 인 출 )  [ ControllerUnit ] - Decode  Opcode,Operend 해석하여  [ ALU ] - Execute  연산,실행  WriteBack - 레지스터 또는 메모리에 저장 | | |  | ISA( Instruction Set Architecture )  명령 집합 구조  각기다른 CPU구조 에서 해당 CPU의 구조에  맞게 실행될 수 있게 적절히 ISA규칙에 맞는  기계어로 번역된다  즉 ISA는 CPU가 어떤구조의 기계어를 이해할 수 있는지 판단하여 코드를 컴파일 하게되면 해당 CPU에 맞는 기계어로 번역된다. | | | 6502 명령어 테이블  각 CPU에 해당하는 명령어 테이블이 존재 | | | | 바이너리 코드를 명령테이블에 따라 CPU의 ISA규치에맞는 언어로 변환한 과정 ( 디코딩 ) | | | |  | |  | |  | | 6502 CPU  레지스터 6개 ( 각 8비트 )  8비트  명령어( opcode )수 56개 | |  | | 6502 CPU 사양 | |  | | 6502 핀 데이터시트  데이터 핀 : D0 ~ D7 ( 8비트 )  주소 핀 : A0 ~ A15 ( 16비트 )  전원 +극 : VDD  접지( Ground ) : VSS  RDY ( Ready ) Cpu사용 준비가 되었다는 걸 의미  준비가 되었다면 항상 1( High )로 연결  IRQB ( Interrupt Request )  외부장치의 인터럽트 요청을 수행한다.  NMIB ( Non - Maskable Interrupt )  하드웨어의 비정상적인 종료같은 예외상황에 대한 인터럽트 요청을 수행한다.  RESB ( Reset )  CPU의 재부팅(리셋)을 한다.  재부팅시 프로그램의 시작 주소(리셋벡터)인  fffc 와 fffd 의 주소를 읽어서 jump하여 실행한다  SOB ( Set Overflow )  연산 결과가 해당 메모리 비트를 초과하여  오버플로우가 발생하면 프로그램 을 멈추게 할지 설정한다 [ 오버플로우 플래그 설정 ] 이라고 한다  PHI2 ( Phase 2 In )  클럭의 입력 신호를 받아서  cpu의 명령 사이클을 실행한다  BE ( Bus Enable )  데이터교환을 할 수있는 데이터버스 / 주소버스 를 활성화 한다.  RWB ( Read Writh )  BE핀이 활성화가 된상태에서  메모리의 데이터를 읽어오고,  메모리에 저장하는 역활 또는 외부장치와의 통신 | |  | | 클럭 (오실레이터)  1초에 CPU의 명령사이클 실행 횟수를 정한다.  1헤르츠( Hz ) =1초에 1번씩 실행  10000MHz => 1초에 100만번 실행 | |  | | VCC 전원  GND 접지  OutPut  Hz에 해당하는 횟수만큼 Low 과 High  신호를 보낸다 | |  | | 음극 에서 전자가 발생하고 - => +  양극 에서 전류가 발생한다. + => - | |  | | ROM ( Read Only Memory )  전원이 꺼져도 데이터가 유지되는대  읽기전용 이므로 한번 데이터 입력을 해놓으면  수정이 불가하다.  EROM( CY62256 X8비트 )  ,EEROM( AT28256 X8비트)  요즘은 데이터 덮어쓰는 방식으로 데이터 수정 가능 | |  | | ROM 핀 데이터시트  WEB ( Write Enable)  데이터를 쓰거나 변경하는 것을 허용 한다  OEB( Output Enable )  데이터 출력을 활성화.  CEB( Chip Enable )  Rom의 활성화.  Cput의 15번핀과 연결하면 클럭에 의해 명령사이클이 돌때 프로그램의 주소벡터를 참조하여 실행한다.  A14  롬은15핀 CPU는16핀 인대  보편적으로 롬의 어드레스 최상위핀은 주소벡터이다 Hight면 Cput가 실행하는 프로그램의 실행위치 주소 이다. Low일 경우 Ram을 사용 한다. | |  | | RAM ( Random Access Memory ) | |  | | Ram 핀 데이터시트  OEB  Cpu의 요청이 있을떄 데이터를 가져올라면  활성화가 되어 있어야겠다  WEB  Cpu가 Ram에 데이터를 저장하려할때  활성화가 되어 있어야겠다  I/O  데이터를 주고 받고 하는 핀  OEB WEB의 활성화에 따라  Cpu의 RWB 를 Ram의 WEB와 버스 연결  Cpu의 최상위 어드레스핀이 0이면 Ram을 사용한다 | |  | | - Binary Code -  (OpCode)operation code / Operand  실행 코드 / 피연산자  - 어셈블리어 -  니모닉 / 피연산자  니모닉 + 주소지정 모드 = OpCode | |  | | 주소지정 기호 $ 의 피연산자는 2바이트를 차지한다  직접대입 하는 상수 기호 # 의 피연산자는 1바이트를 차지한다 | |  | | 니모닉+주소지정 모드(Addressing mode) = OpCode  니모닉은 주소지정 모드에 따라  OpCode가 결정된다 | |  | | - A 모드 -  ALU의 계산결과가 들어가는 레지스터  1 바이트  실행사이클 2 | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | | | |
|  | | |