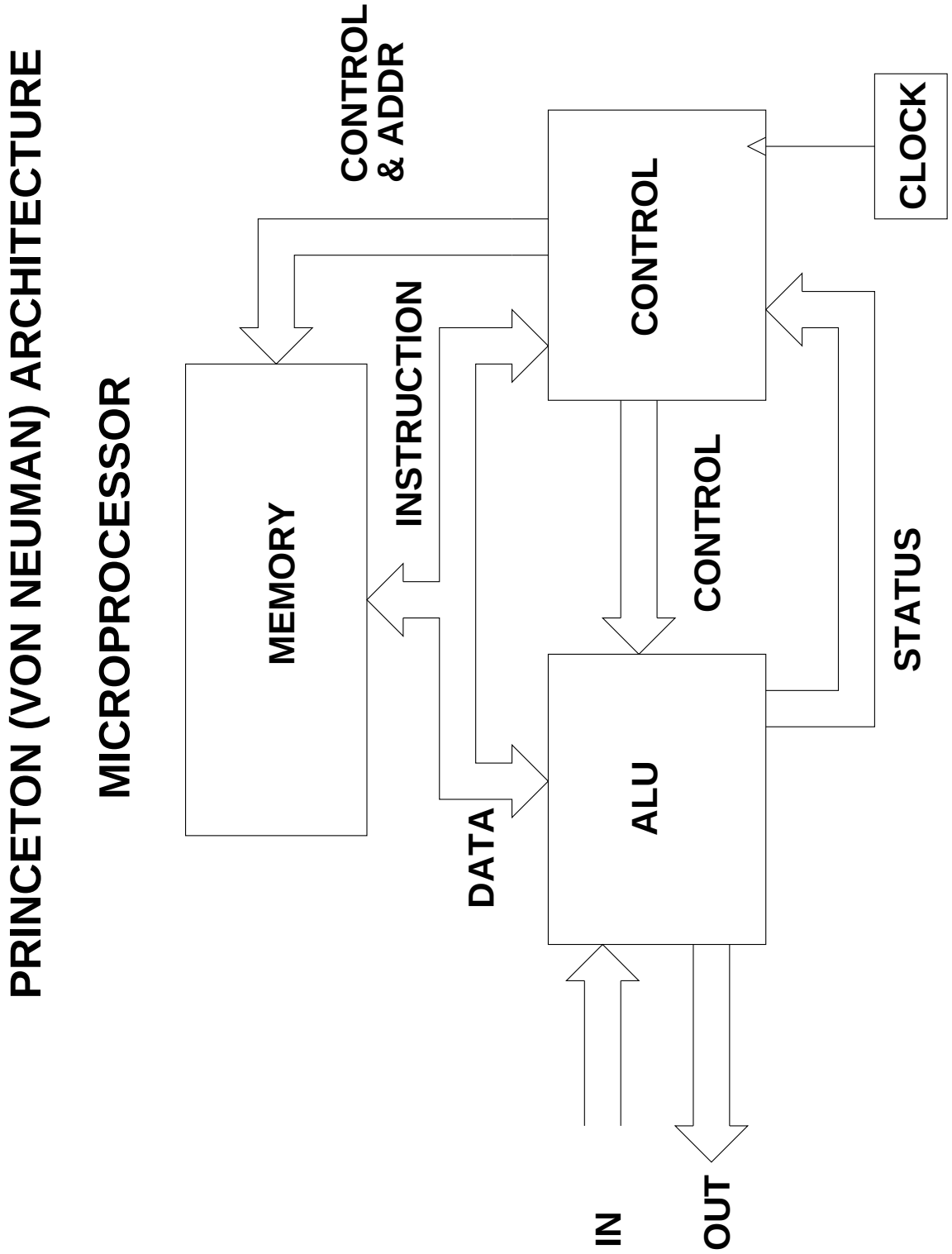
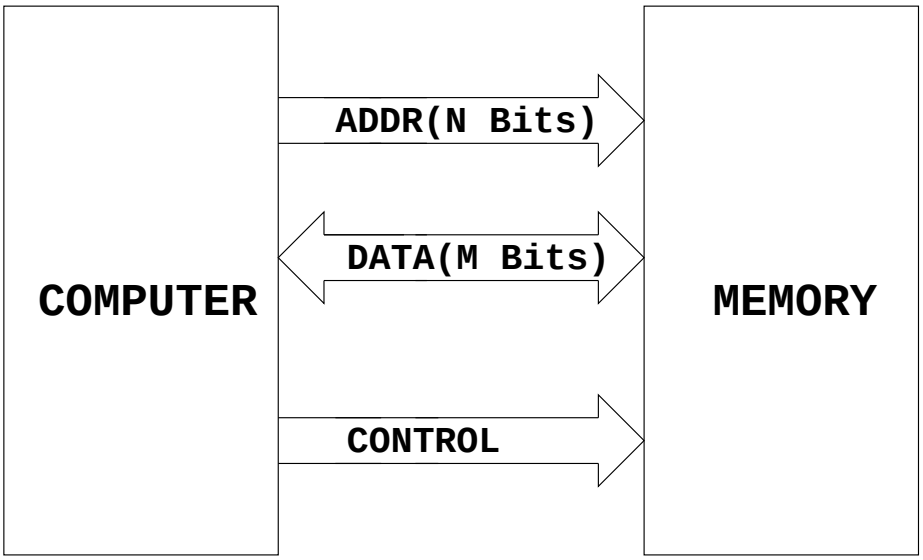

PRINCETON (VON NEUMAN) ARCHITECTURE
MICROPROCESSOR
CONTROL & ADDR
CLOCK
CONTROL
MEMORY
INSTRUCTION
CONTROL
STATUS
ALU
DATA
IN
OUT

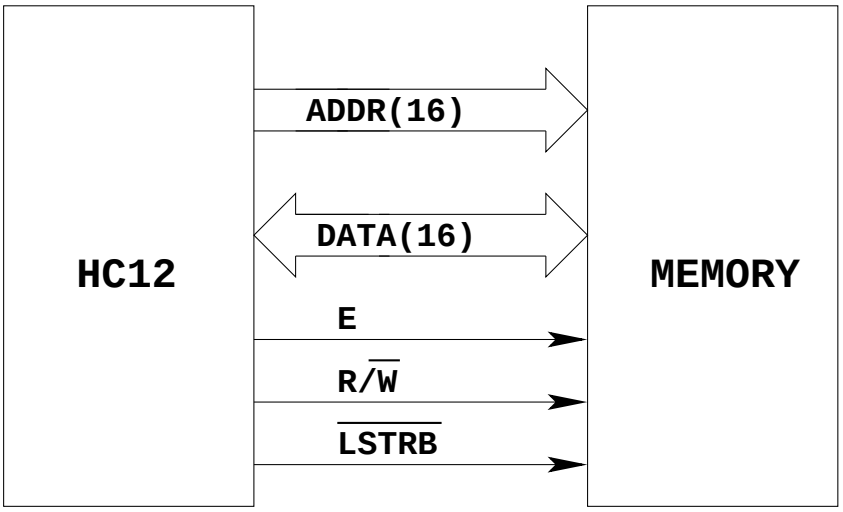


Computer with N bit address bus can access 2^N bytes of data

Computer with M bit data bus can access M bits of data in one memory cycle

Value on address bus tells memory which location computer wants to read (write)

Control lines tell memory when computer wants to read (write) data, and if access is read or write



HC 12 has 16 bit address bus – can access 65536 bytes

1024 bytes = 1 kB

65536 bytes = 64 kB

HC12 has 16 bit data bus – can access 16 bits (2 bytes) at a time

For example, the instruction LDX $0900 will read the two bytes at address $0900 and $0901

Sometimes HC12 only accesses one byte -- e.g., LDAA $0900
The HC12 accesses only the byte at address $0900

R/$\overline{W}$ tells memory if HC12 is reading or writing

R/$\overline{W}$ high => read
R/$\overline{W}$ low => write

E tells memory when HC12 is reading (writing) -- synchronizes data accesses

$\overline{LSTRB}$ tells memory if HC12 accessing one or two bytes

Address, Data and Control Buses

- A microprocessor system uses address, data and control buses to communicate with external memory and memory-mapped peripherals
- The address bus determines which memory location to access
- The control bus specifies whether the memory cycle is a read (into microprocessor) or a write (out of microprocessor) cycle, and specifies timing information for the cycle
- The data bus contains the data being transfered during the memory cycle
- For example, consider the following simple HC12 program, which continuously increments the contents of address `0x0400`:

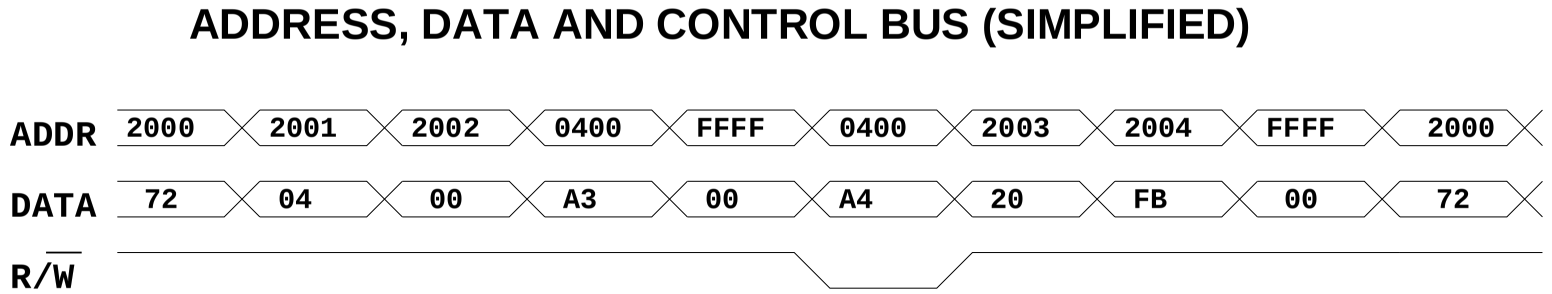
```
        org     0x2000

loop:   inc     0x0400
        bra     loop
```

  - The program is stored in memory starting at memory location `0x2000`
  - The HC12 Program Counter starts at address `0x2000`
  - The HC12 reads the first instruction, `inc 0x0400`, located in address `0x2000` through `0x2002`
  - The HC12 then reads the contents of memory location `0x0400`, takes an internal memory cycle to increment the value, then writes the new value out to address `0x0400`
  - The HC12 then reads the next instruction, `bra 0x2000`
  - The HC12 takes one memroy cycle to load the program counter with the new value of `0x2000`, and to clear its internal pipeline, then reads the instruction at `0x2000` to figure out what to do next

The HC12 address, data and control buses (simplified)

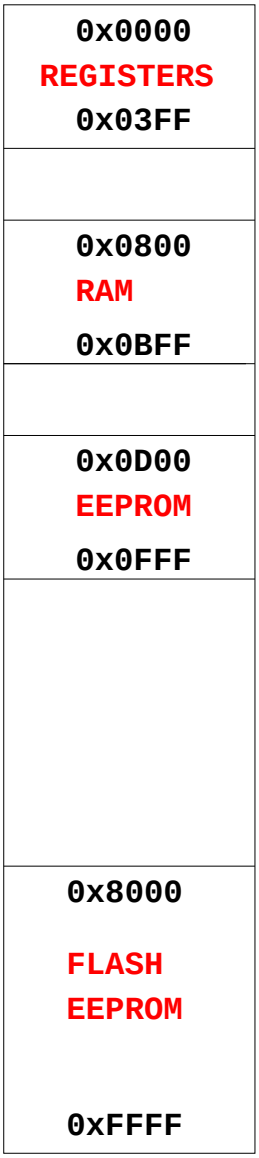
- Note: The following diagram assumes that the HC12 accesses one byte at a time
- The HC12 actually accesses two bytes (16 bits) at a time, when it can
- What actually occurs on the HC12 bus is a little more complicated than what is shown below




```
                .org   0x2000        2000:  72
                                    2001:  04
loop:           inc    0x0400       2002:  00
                                    2003:  20
                bra    loop         2004:  FB
```

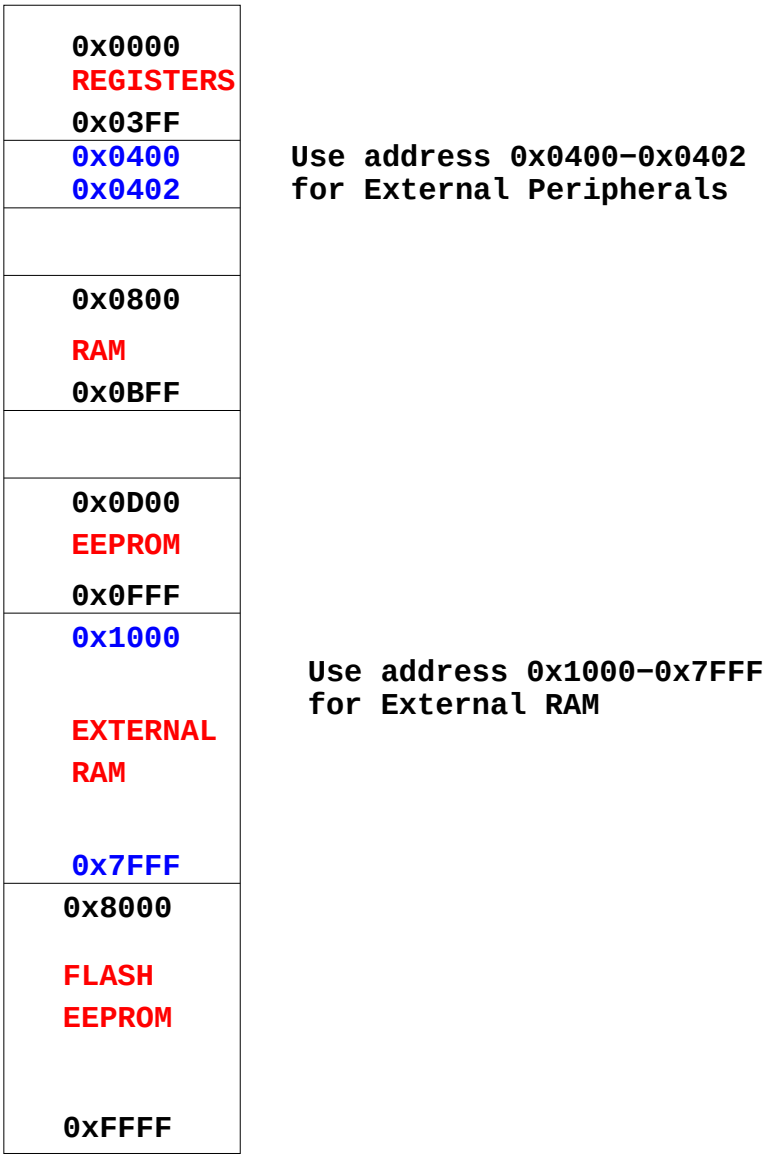
The HC12 Memory Map

- The HC12 has address regions occupied by internal memory and peripherals
- A diagram showing which address regions are used is called a memory map
- Here is a memory map of the HC12B32 with no added memory or peripherals



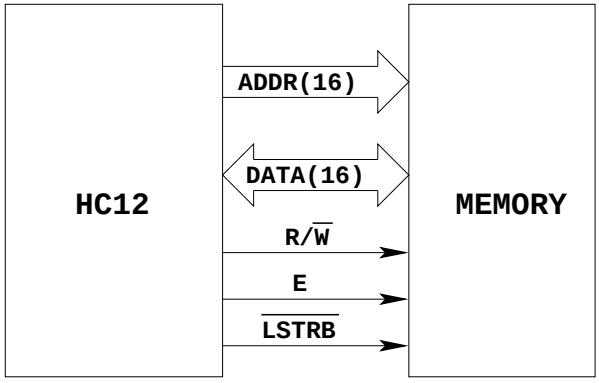
The Expanded HC12 Memory Map

- We will add external memory to the HC12
- Here is a memory map of the HC12B32 with the memory and peripherals we will add



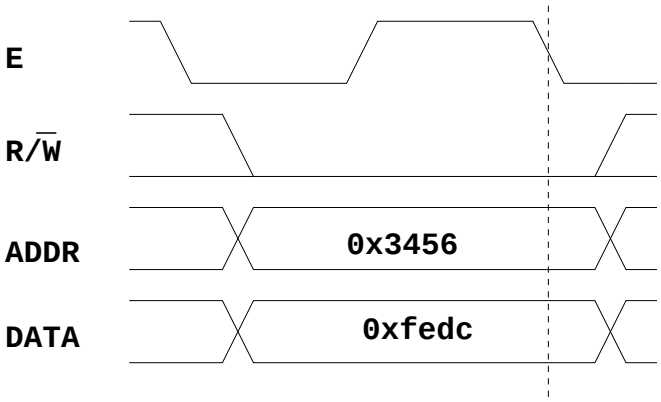
Simplified HC12 Write Cycle

- When the HC12 writes data to memory it does the following:
  - It puts the address it wants to write to on the address bus (when E-clock goes low)
  - It puts the data it wants to write onto the data bus
  - It brings the Read/Write ($R/\overline{W}$) line low to indicate a write
  - The HC12 expects the external device at the given address will latch the data into its registers data on the falling edge of the E-clock



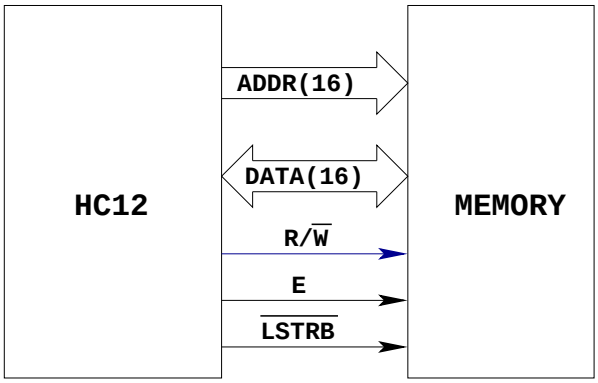
WRITE: HC12 puts address on address bus
puts data on data bus
brings R/$\overline{W}$ low
Memory latches data on falling edge of E clock

Example: Write 0xfedc to address 0x3456 & 3457



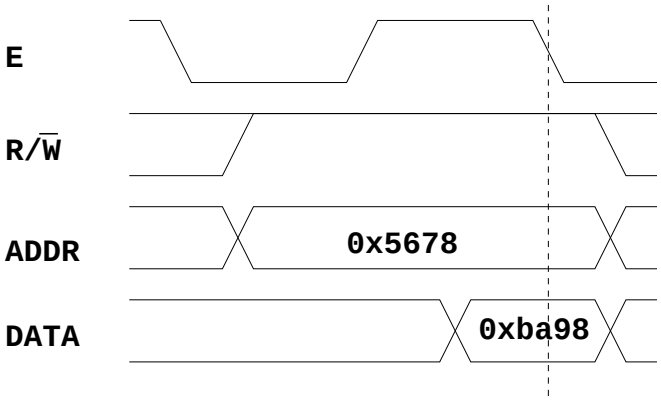
Simplified HC12 Read Cycle

- When the HC12 reads data from memory it does the following:
  - It puts the address it wants to read from on the address bus (when E-clock goes low)
  - It brings the Read/Write ($R/\overline{W}$) line high to indicate a read
  - The HC12 expects the external device at the given address will put data on the data bus
  - On the falling edge of the E-clock, the HC12 latches the data into its internal registerl



READ: **HC12 puts address on address bus**
brings R/$\overline{W}$ high
Memory puts data on data bus
HC12 latches data on falling edge of E clock

Example: Read from address 0x5678 & 0x5679



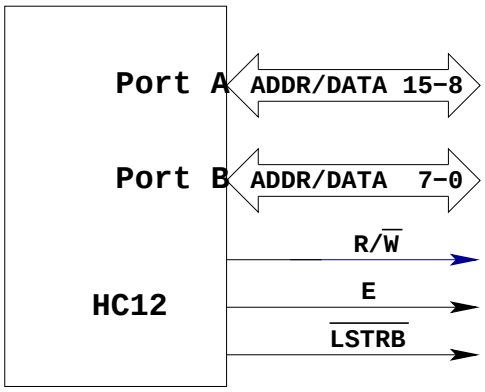
The Real HC12B32 Bus

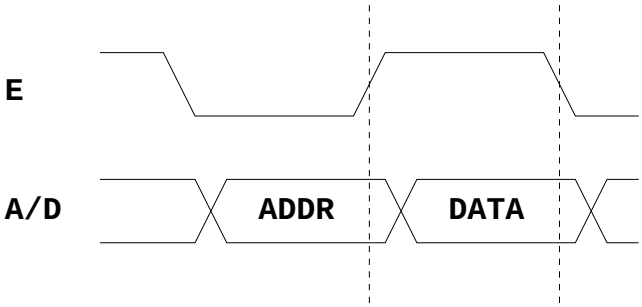
- Up to now we have been using the HC12 in Single Chip Mode
  - In Single Chip Mode the HC12 does not have an external address/data bus
- The HC12 can be run in Expanded Mode
  - In Expanded Mode the HC12 does have an external address/data bus
- Things are a little more complicated on the real HC12B32 bus than shown in the simplified diagrams above
- The HC12B32 has a multiplexed address/data bus
- The HC12B32 sometimes accesses a single byte on a memory cycle, and it sometimes access two bytes on a memory cycle

The Multiplexed Address/Data Bus

- The HC12B32 has a limited number of pins it can use
- To have full 16-bit address bus and a full 16-bit data bus the HC12B32 would need to use 32 extra pins (in addition to several pins used for the control bus)
- To save pin count Motorola uses the same set of pins for several purposes
- When put into expanded mode, the HC12 uses the pins normally used for Ports A and B for its mulitplexed address and data bus
  - When running in expanded mode you can no longer use Ports A and B as general purpose I/O lines
- The HC12 uses the same sixteen line of Ports A and B for both address and data
- When the E-clock is low the sixteen lines AD15-0 are used for address
- When the E-clock is high the sixteen lines AD15-0 are used for data

The Multiplexed Address/Data Bus





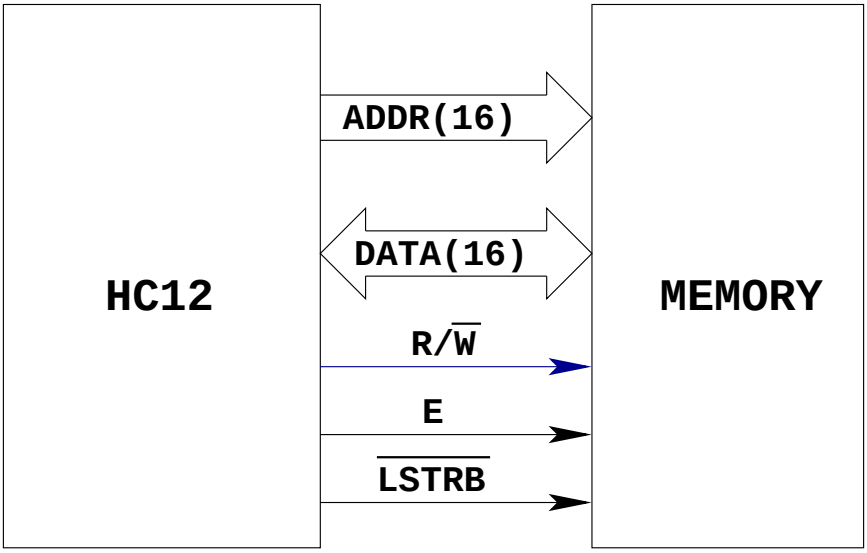
In expanded mode, the HC12 uses Ports A and B for mulitplexed address and data.

When E clock is low, Ports A and B are used for address.

When E clock is high, Ports A and B are used for data.

In expanded mode, you cannot use Ports A and B for general purpose I/O.

The Multiplexed Address/Data Bus



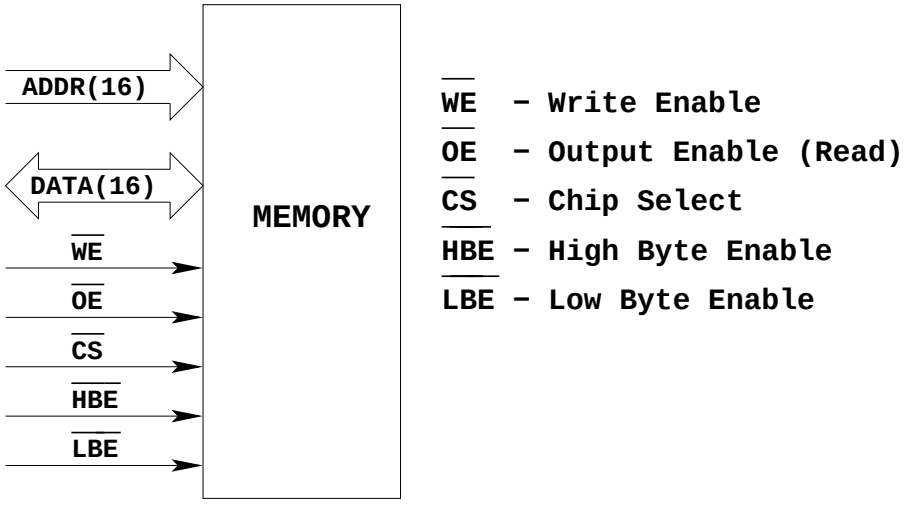
HC12 has 16−bit address and 16−bit data bus

Requires 35 bits

Not enough pins on HC12 to allocate 35 pins
for buses and pins for all other functio

Memory Chip Interface

- Memory chips need separate address and data bus
  - Need way to de-multiplex address and data lines from HC12
- Memory chips need different control lines than the HC12 supplies
- These control lines are:
  - Chip Select – goes low when the HC12 is accessing memory chip
  - Write Enable – goes low when the HC12 is writing to memory
  - Output Enable – goes low when the HC12 is reading from memory
  - High Byte Enable – goes low when the HC12 is accessing the High Byte (Odd Address) of memory
  - Low Byte Enable – goes low when the HC12 is accessing the Low Byte (Even Address) of memory

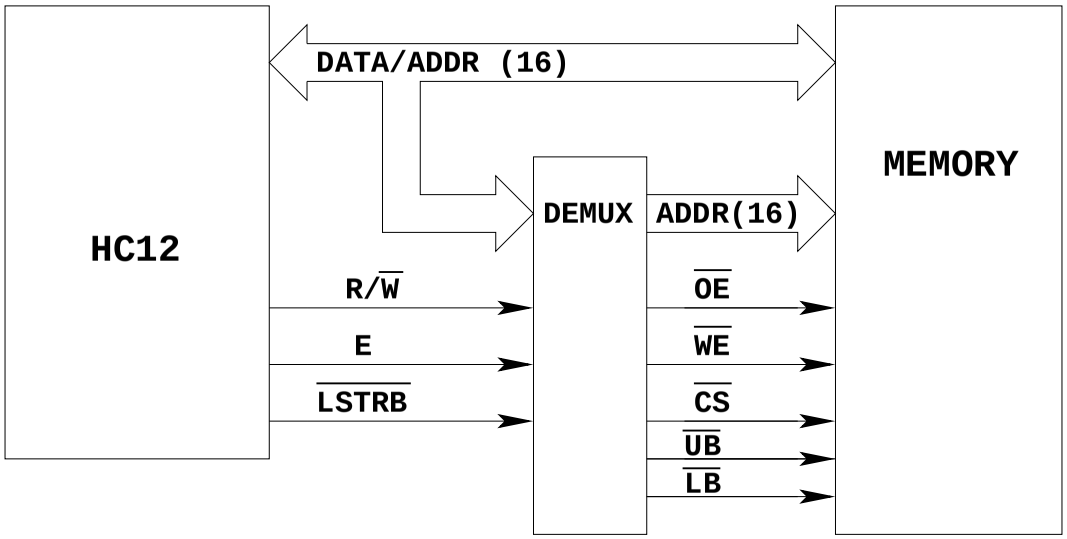


Memory needs separate address and data busses

Need way to separate address and data

The Multiplexed Address/Data Bus

- To talk to memory chip we will need to build a demultiplexer between the HC12 and the memory chip



$\overline{OE}$ – Output Enable (Read)
$\overline{WE}$ – Write Enable
$\overline{CE}$ – Chip Enable
$\overline{UB}$ – Upper Byte Enable
$\overline{LB}$ – Lower Byte Enable

HC12 has 16-bit address and 16-bit data buses

Requires 35 bits

Not enough pins on HC12 to allocate 35 pins
for buses and pins for all other functions

Solution: multiplex address and data buses

16-bit Bus: While E low, bus supplies address
While E high, bus supplies data