Chapter 9

Hardware Architecture of 68000

Expected Outcomes
- Describe the internal architecture of 68000
- Describe general specification of 68000 microprocessor
- Outline the processor’s control signals name and functions
- Sketch the general timing signal for read and write operation
Product Numbering System for the 68xxx Family (Motorola Product)

- **Status Product**
  - MC - Fully qualified
  - XC - Partial Qualified
  - PC - Product Engineering

- **Part/Module Modifier**
  - EC - Embedded Controller
  - LC - Low cost, no FPU
  - BLANK - Full CPU
  - HC - HCMOS
  - SEC - Static Embedded

- **Product Designator**
  - 000,010,020 etc

- **Die Mask Revision**

- **Voltage**
  - BLANK = 5V
  - V = 3.3V

- **Speed**

- **Packaging Designator**

- **Temperature Range**
  - BLANK = 0 - 70°C
  - B = -40 - 70°C
  - C = -40 - 85°C
Pin Configuration

MC68000/P10

- C4
- C3
- C2
- C1
- C0
- AS
- LDS
- LDS
- R/W
- DTACK
- BG
- BGACK
- BR
- CLK
- GND
- HALT
- RESET
- VIA
- E
- VPA
- BERR
- IPL2
- IPL1
- IPL0
- FC2
- FC1
- FC0
- A1
- A0
- A3
- A4

MC68000

- VCC (2)
- GND (2)
- CLK
- IPL0*
- IPL1*
- IPL2*
- E
- VMA*
- VPA*
- BERR*
- RESET*
- HALT*
- BR*
- BG*
- BGACK*

ADDRESS: A1-A23
DATA: D0-D15

AS*
R/W*
UDS*
LDS*
DTACK*
# 68000 Manufacturer

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple 68000-8</td>
<td>8 MHz, 64-pin side-brazed ceramic DIP</td>
</tr>
<tr>
<td>Hitachi HD68000-8</td>
<td>8 MHz, 64-pin side-brazed ceramic DIP</td>
</tr>
<tr>
<td>Mostek MK68000-8B</td>
<td>8 MHz, 64-pin plastic DIP</td>
</tr>
<tr>
<td>Motorola XC68000L</td>
<td>64-pin side-brazed ceramic DIP</td>
</tr>
<tr>
<td>Rockwell R68000C8</td>
<td>8 MHz, 64-pin side-brazed ceramic DIP</td>
</tr>
<tr>
<td>SGS-Thompson TS68000CP10</td>
<td>10 MHz, 64-pin plastic DIP</td>
</tr>
<tr>
<td>Signetics SCN68000C4164</td>
<td>4 MHz, 64-pin side-brazed ceramic DIP</td>
</tr>
<tr>
<td>Thompson TS68000CFN16</td>
<td>16 MHz, 68 Lead plastic LCC</td>
</tr>
<tr>
<td>Toshiba TMP68HC000P-10</td>
<td>10MHz, 64-pin plastic DIP</td>
</tr>
<tr>
<td>Motorola MC68HC000LC8</td>
<td>8 MHz, 64-pin side-brazed ceramic DIP</td>
</tr>
<tr>
<td>Hitachi HD68000Y10</td>
<td>10 MHz, 68-pin ceramic PGA</td>
</tr>
</tbody>
</table>
**Timing Diagram**

- **Instruction cycle**
  - A complete cycle for 68000 to read and execute an instruction

- **Bus cycle**
  - Time for 68000 to read or write a byte/word from memory

- **Clock cycle (cc)**
  - Time between two consecutive positive edge and equal to clock duration and require 4 cc in each bus cycle

![Timing Diagram](image-url)
68000 Pin Function

VCC (2)
GND (2)
CLK
IPL0*
IPL1*
IPL2*
E
VMA*
VPA*
BERR*
RESET*
HALT*
BR*
BG*
BGACK*

ADDRESS
A1-A23
DATA
D0-D15
AS*
R/W*
UDS*
LDS*
DTACK*
FC0
FC1
FC2

MC68000
68000 Pin Function

Power, Clock, data and address bus

- **VCC and GND**
  - Power supply to 68000 with 2 pin each for VCC and GND
  - 5V ± 5% for VCC with power consumption of 1.5W at 8MHz and current about 0.3A

- **CLK** - Various clock of 4, 6, 8, 10, 12.5, 16.67 and 25 MHz

- **D0-D15** - Bidirectional data bus that are used to transfer data in or out of CPU

- **A1-A23** - Unidirectional address bus with A0 is replaced by UDS* and LDS* to allow byte and word operation
68000 Pin Function

Asynchronous Signal for Hardware Operation

- **AS* (Address Strobe) - output**
  - When AS* goes low, A1 – A23 are valid address

- **R/W* (Read/Write) - output**
  - Determine read (high) or write (low) operation

- **UDS* (Upper Data Strobe) - output**
  - Control the transfer of upper data bus (D8 – D15)

- **LDS* (Lower Data Strobe) - output**
  - Control the transfer of upper data bus (D0 – D7)
68000 Pin Function

**DTACK** (Data Transfer Acknowledge) - **input**
- When CPU receives a low signal, it assumes this is the end of bus cycle
- I/O device must activate the DTACK pin correctly

<table>
<thead>
<tr>
<th>Signal</th>
<th>Function (when asserted)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS</td>
<td>Output signal indicating valid memory address available on address bus</td>
</tr>
<tr>
<td>R/W</td>
<td>Output signal indicating read cycle when HIGH and write cycle when LOW</td>
</tr>
<tr>
<td>UDS</td>
<td>Together with LDS signal, output an encoded signal based on internal state of A0 address line</td>
</tr>
<tr>
<td>LDS</td>
<td>Together with UDS signal, output an encoded signal based on internal state of A0 address line</td>
</tr>
<tr>
<td>DTACK</td>
<td>Input signal indicating data transfer during read/write cycle is completed</td>
</tr>
</tbody>
</table>
Due to the data bus size of 16-bit, memory map is consisted of two byte.

- **UDS** and **LDS** are used to access even and odd location respectively.
- To access a byte of odd address, **UDS** = 1, **LDS** = 0.
- To access a byte of even address, **UDS** = 0, **LDS** = 1.
- To access a word, **UDS** = 0, **LDS** = 0.
Byte Addressing

<table>
<thead>
<tr>
<th>UDS*</th>
<th>LDS*</th>
<th>R/W*</th>
<th>D8 - D15</th>
<th>D0 - D7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>-</td>
<td>Illegal data</td>
<td>Illegal data</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Bit 8 - 15</td>
<td>Bit 0 - 7</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Illegal data</td>
<td>Bit 0 - 7</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Bit 8 - 15</td>
<td>Illegal data</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Bit 8 - 15</td>
<td>Bit 0 - 7</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Illegal data</td>
<td>Bit 0 - 7</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Bit 8 - 15</td>
<td>Illegal data</td>
</tr>
</tbody>
</table>
Timing Diagram - READ

**Device address**
1) Set R/W* pin to read mode
2) Place code function into FC0-FC2
3) Place address into A1-A23
4) Activate AS*
5) Activate UDS* and LDS*

**Output data**
1) Decode address
2) Place data into D0-D15
3) Activate DTACK*

**Terminate data transfer**
1) Accept data
2) Deactivate UDS* and LDS*
3) Deactivate AS*

**Terminate cycle**
1) Disable data
2) Deactivate DTACK*

**initiate next cycle**
Timing Diagram - READ

CLK
A1 - A23
valid address
R/W*
AS*
UDS*, LDS*
DTACK*
D0- D15 valid data
Timing Diagram - WRITE

Device address
1) Place code function into FC0-FC2
2) Place address into A1-A23
3) Activate AS*
4) Activate write pin (R/W*)
5) Place data into D0-D15
6) Activate UDS* and LDS*

Output data
1) Decode address
2) Accept data through D0-D15
3) Activate DTACK*

Terminate data transfer
1) Deactivate UDS* and LDS*
2) Deactivate AS*
3) Deactivate data
4) Deactivate write (R/W*)

Terminate cycle
1) Deactivate DTACK*

Initiate next cycle
Timing Diagram - WRITE

CLK
A1 - A23 valid address
R/W*
UDS*, LDS*
AS*
DTACK*
D0- D15 valid data
Interfacing With 6800 Peripheral

- Most cases, 68000 uses 6800 peripherals as interface devices due to low cost
- Three control signals are used to ensure functionality of the system
  - **E** (Enable clock) – output
    - Generate timing signal for 6800 peripherals and is derived from the 68000’s clock by dividing it by 10 with resulting waveform having 40% duty cycle
  - **VPA* (Valid Peripheral Address)** – input
    - Inform 68000 that it has addressed a 6800 peripheral and that the data transfer should be synchronized with **E** clock

![Clock Waveform Diagram](image-url)
Interfacing With 6800 Peripheral

**VMA*** (Valid Memory Address) – output

- Goes low when the processor synchronizes with E clock allowing data transfer

<table>
<thead>
<tr>
<th>Signal</th>
<th>Function (when asserted)</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>Output a synchronous timing (clock) for proper information transfer between the microprocessor and the M6800 peripheral devices</td>
</tr>
<tr>
<td>VPA</td>
<td>inform the microprocessor that an M6800 peripheral device has been addressed and data transfer to be synchronized with E clock</td>
</tr>
<tr>
<td>VMA</td>
<td>inform M6800 peripheral device that it has been addressed by the address bus and data transfer is synchronized to E clock</td>
</tr>
</tbody>
</table>
Interfacing With 6800 Peripheral

Interface with peripheral devices involves $E$, $\text{VMA}^*$, and $\text{VPA}$ pins to ensure proper functionality of the system as 6800 devices operate in synchronous mode.
Interfacing With 6800 Peripheral

1) CPU initiates read or write cycle

Define 6800 cycle
1) External hardware activate VPA*

Synchronize with E clock
1) CPU waits until E goes low
2) CPU activate VMA*

Data Transfer
1) Peripheral waits until E is active before transferring data

Terminate cycle
1) CPU waits until E goes low
2) CPU disables VMA*
3) CPU disable AS*, UDS* and LDS*

initiate next cycle
Bus Arbitration

- This bus signals are used in multiprocessor system and DMA.
- It allows external device to take over control of the bus (master) and place 68000 in wait state.

**BR* (Bus Request) – input**
- The requesting device requests use of 68000 buses by activate this pin (goes low).

**BG* (Bus Grant) – output**
- 68000 respond by activate this pin (goes low) and it will release control of its buses at the end of the current cycle.

**BGACK* (Bus Grant Acknowledge) – input**
- The requesting device informs to the 68000 of bus control and it must wait until the bus cycle is terminated before issue this acknowledgement.
Bus Arbitration

This arbitration signals are required to allow more than one processor to control buses.
Exception Processing

- 68000 microprocessor operates in two modes; user execution mode and supervisor execution mode
- In user execution mode, the microprocessor operates in normal processing environment suitable for user’s application programs
- Supervisor mode provides programmers with full access to the microprocessor’s instructions and status register
- The mode is determined by the thirteenth bit (S-bit) in the status register. Setting the S-bit in the status register puts the microprocessor in the supervisor mode
- Exception is one of the three processing states of a microprocessor. This is the processing state where microprocessor handles all internally or externally generated exceptions
- Examples of exception are external hardware interrupts, external hardware RESET, internal RESET/TRAP instructions, tracing, bus error and execution errors
## Exception Vector

<table>
<thead>
<tr>
<th>Vector Number</th>
<th>Address ($)</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>000</td>
<td>RESET: Initial SSP</td>
</tr>
<tr>
<td>1</td>
<td>004</td>
<td>RESET: Initial PC</td>
</tr>
<tr>
<td>2</td>
<td>008</td>
<td>Bus Error</td>
</tr>
<tr>
<td>3</td>
<td>00C</td>
<td>Address Error</td>
</tr>
<tr>
<td>4</td>
<td>010</td>
<td>Illegal Instruction</td>
</tr>
<tr>
<td>5</td>
<td>014</td>
<td>Zero Divide</td>
</tr>
<tr>
<td>6</td>
<td>018</td>
<td>CHK Instruction</td>
</tr>
<tr>
<td>7</td>
<td>01C</td>
<td>TRAPV Instruction</td>
</tr>
<tr>
<td>8</td>
<td>020</td>
<td>Privilege Violation</td>
</tr>
<tr>
<td>9</td>
<td>024</td>
<td>Trace</td>
</tr>
<tr>
<td>10</td>
<td>028</td>
<td>Line 1010 Emulator</td>
</tr>
<tr>
<td>11</td>
<td>02C</td>
<td>Line 1111 Emulator</td>
</tr>
<tr>
<td>12-14</td>
<td>030-038</td>
<td>Reserved</td>
</tr>
<tr>
<td>15</td>
<td>03C</td>
<td>Uninitialized Interrupt Vector</td>
</tr>
<tr>
<td>16-23</td>
<td>040-05C</td>
<td>Reserved</td>
</tr>
<tr>
<td>24</td>
<td>060</td>
<td>Spurious Interrupt</td>
</tr>
<tr>
<td>25-31</td>
<td>064-07C</td>
<td>Interrupt Auto-vector Level 1-7</td>
</tr>
<tr>
<td>32-47</td>
<td>080-0BC</td>
<td>TRAP Instruction Vectors</td>
</tr>
<tr>
<td>48-63</td>
<td>0C0-0FF</td>
<td>Reserved</td>
</tr>
<tr>
<td>64-255</td>
<td>100-3FC</td>
<td>192 User Interrupt Vectors</td>
</tr>
</tbody>
</table>
Self-Test

Exercise

Explain the difference between synchronous and asynchronous data bus transfer and explain the bus signals used in each method.

Exercise

State the number of data and address bus for 68000. Explain why the A0 pin is not part of 68000.

Exercise

State the logic of UD$*, LD$*, AS$*, A13 and R/W$* when

(i) a byte write to address $4000
(ii) a byte read from address $5678
(iii) a word write to address $6542
Self-Test

Exercise

Explain the importance of DTACK pin in asynchronous data bus transfer

Exercise

Briefly explain the use of wait states in memory access

Exercise

State the bus direction for each pin

(i) DTACK*
(ii) D8
(iii) A6
(iv) R/W*
(v) BGACK*
(vi) VPA*
(vii) E
(viii) UDS*