Hardware Details of the 68000
- CPU Pin Descriptions
- System Timing Diagrams

CPU Pin Descriptions

Figure 7.1: 68000 CPU input and output signals

Unless otherwise specified, all materials and diagrams are adapted from the following sources:
**V\text{CC}, GND, CLK**

- Processor power and clock inputs.
- 2 pins each for VCC and GND
- Both must be connected

**VCC:**
- 8MHz, 5V ± 5%, 1.5 watts

**CLK:**
- Max $t_{\text{rise}}$ and $t_{\text{fall}}$ are 10ns (all versions except 5ns for 12.5MHz and 16MHz)
- TTL-compatible with 50 percent duty cycle.
FC₀, FC₁, and FC₂
- FC₀, FC₁, and FC₂ are function code outputs
- Informs external circuitry of current internal processing state of the 68000.
- Only valid when AS is active.

Table 7.1: Function code outputs (pg 241) [1]

<table>
<thead>
<tr>
<th>FC₂</th>
<th>FC₁</th>
<th>FC₀</th>
<th>Cycle type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Reserved*</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>User data</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>User program</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Reserved*</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Reserved*</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Supervisor data</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Supervisor program</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Interrupt acknowledge</td>
</tr>
</tbody>
</table>

*By Motorola, for future use.

- Frequently used to restrict memory accesses by connecting the FC pins to the memory address decoding circuitry.
Figure 7.3: Interrupt acknowledge cycle decoder (pg 242) [1]
\[ E, \overline{VMA}, \text{ and } \overline{VPA} \]

- Provides the capability to control older 6800 peripherals.
- \(E\) clock, \(\overline{VMA}\) (valid memory address), and \(\overline{VPA}\) (valid peripheral address)

\(E\) clock:
- Generates proper timing signals for 6800 peripherals.
- \(1/10\)th of 68000’s clock frequency, with 40% duty cycle (high for 4 CLK cycles, low for 6 CLK cycles)
Sequence of events during 6800 peripheral access:
- The 68000 places address of 6800 peripheral on A₁ through A₂₃.
- $VPA$ input is activated (low) to request synchronization of 68000
- 68000 synchronizes with the E clock, then activates $VMA$ (outputs a low).
- Data transfer takes place.
Figure 7.6(b): 6800 peripheral interface

Unless otherwise specified, all materials and diagrams are adapted from the following sources:
RESET, HALT, and BERR

- Provides system control.
- RESET, HALT, and BERR (buss error) are panic buttons of the 68000
- External circuitry pulls BERR low to indicate to the 68000 that an error has occurred during execution of current bus cycle.
- RESET and HALT lines are bi-directional

Sequence of events when bus error occurs:

- If HALT is not asserted when bus error occurs:
  - The 68000 will terminate the current failed cycle and start bus error exception processing.
If $HALT$ was asserted before or at the same time as $BERR$:

- The 68000 will terminate the current failed cycle, and enter a “do nothing” state.
- Once the $HALT$ line is deactivated, the processor will rerun the previous cycle.
- Note that $BERR$ must be deactivated at least one clock cycle before $HALT$ is deactivated.
Example (Power On Hardware Reset):

- RESET and HALT both act as inputs
- After power on, RESET and HALT are taken low for at least 100ms
- This stabilizes V_{CC} and results in total processor reset

Example (Normal Execution Hardware Reset):

- RESET and HALT both act as inputs
- Asserting RESET and HALT at least 10 clock cycles resets the 68000
Figure 7.9: Power-on \textit{RESET} and \textit{HALT} timing (pg 246) [1]

Unless otherwise specified, all materials and diagrams are adapted from the following sources:

Example (Normal Execution Peripheral Soft Reset):

- **RESET** acts as an output
- The RESET instruction causes the 68000 to output a low level on **RESET** for 124 clock cycles.
- This resets all external circuitry connected to **RESET** without affecting the state of the processor.
Figure 7.7: Power-on \textit{RESET} and \textit{HALT} timing (pg 245) [1]
Example (External Peripheral Halt):

- $HALT$ acts as an input
- $HALT$ may be driven low at any time by an external device.
- The processor then completes the current bus cycle and halts:
  - All tristate signals are set to high-impedance
  - All control signals are inactive.
- Execution returns to normal when $HALT$ is deactivated
Figure 7.8: Tristate signal levels

Unless otherwise specified, all materials and diagrams are adapted from the following sources:

\[ \overline{IPL_0}, \overline{IPL_1} \text{ and } \overline{IPL_2} \]

- Provides interrupt control/request
- \( \overline{IPL_0}, \overline{IPL_1} \text{ and } \overline{IPL_2} \) are used by external circuitry to send an interrupt request to the 68000.
- Vectored and auto-vectored interrupts may be requested.
<table>
<thead>
<tr>
<th>$\overline{IPL_2}$</th>
<th>$\overline{IPL_1}$</th>
<th>$\overline{IPL_0}$</th>
<th>Interrupt level*</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0 (Lowest, none)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7 (Highest, nonmaskable)</td>
</tr>
</tbody>
</table>

*Note the inversion of the binary bits needed on $\overline{IPL_2}$-$\overline{IPL_0}$.

Figure 7.2: Interrupt level encoding
Figure 7.10: Generating a level-7 interrupt with a pushbutton (pg. 247) [1]
**BR, BG and BGACK**

- Provides bus arbitration control (use to place 68k in a wait state while hardware connected to the bus)
- **BR** (bus request), **BG** (bus grant), and **BGACK** (bus grant acknowledge) are used when a device (DMA controller or another processor) is taking over control of the bus
Sequence of events for bus arbitration control:
- Requesting device (bus master) activate \( BR \) low to request the 68k system bus.
- 68k will respond to bus master by asserting \( BG \) low to indicate its willingness to release control of the bus at the end of the current cycle.
- To take control, the new bus master asserts \( BGACK \).
- When the new bus master wishes to relinquish control of the bus, it does so by negating \( BGACK \).
There are 4 conditions that must be met before the new bus master may activate $BGACK$:

- $BG$ is active
- $AS$ is inactive
- $DTACK$ is inactive
- $BGACK$ is inactive

Provides asynchronous bus control (for proper operation of external hardware)

- All lines are outputs except $DTACK$ (input)

- $AS$ (address strobe): to indicate that a valid memory address exists on the address bus.
- \( R/W \) (read/write): determine whether the current cycle is a read or write.
- \( UDS \) (upper data strobe)
- \( LDS \) (lower data strobe)

  - To transfer 8 bits (1 byte) of data: either \( UDS \) or \( LDS \) asserted
  - To transfer 16 bits: both \( UDS \) and \( LDS \) asserted low

- \( DTACK \) (data transfer acknowledge)

  - When \( DTACK \) is asserted by external hardware, the 68000 recognizes that the current bus cycle can be completed