3 Memory Organization

The memory space of the C167 is configured in a “Von Neumann” architecture. This means that code and data are accessed within the same linear address space. All of the physically separated memory areas, including internal ROM/Flash (where integrated), internal RAM, the internal Special Function Register Areas (SFRs and ESFRs), the address areas for integrated XBUS peripherals (e.g., XRAM or CAN module) and external memory are mapped into one common address space.

The C167 provides a total addressable memory space of 16 MBytes. This address space is arranged as 256 segments of 64 KBytes each, and each segment is again subdivided into four data pages of 16 KBytes each (see figure below).

Figure 3-1
Memory Areas and Address Space
Most internal memory areas are mapped into segment 0, the system segment. The upper 4 KByte of segment 0 (00'F000H...00'FFFFH) hold the Internal RAM and Special Function Register Areas (SFR and ESFR). The lower 32 KByte of segment 0 (00'0000H...00'7FFFH) may be occupied by a part of the on-chip ROM or Flash memory and is called the Internal ROM area. This ROM area can be remapped to segment 1 (01'0000H...01'7FFFH), to enable external memory access in the lower half of segment 0, or the internal ROM may be disabled at all.

Code and data may be stored in any part of the internal memory areas, except for the SFR blocks, which may be used for control/data, but not for instructions.

**Note:** Accesses to the internal ROM area on ROMless devices will produce unpredictable results.

Bytes are stored at even or odd byte addresses. Words are stored in ascending memory locations with the low byte at an even byte address being followed by the high byte at the next odd byte address. Double words (code only) are stored in ascending memory locations as two subsequent words. Single bits are always stored in the specified bit position at a word address. Bit position 0 is the least significant bit of the byte at an even byte address, and bit position 15 is the most significant bit of the byte at the next odd byte address. Bit addressing is supported for a part of the Special Function Registers, a part of the internal RAM and for the General Purpose Registers.

**Figure 3-2**
Storage of Words, Byte and Bits in a Byte Organized Memory

**Note:** Byte units forming a single word or a double word must always be stored within the same physical (internal, external, ROM, RAM) and organizational (page, segment) memory area.
3.1 Internal ROM

The C167 may reserve an address area of variable size (depending on the version) for on-chip mask-programmable ROM (organized as $X \cdot 32$) or Flash memory. The lower 32 KByte of the on-chip ROM/Flash are referred to as “Internal ROM Area”. Internal ROM accesses are globally enabled or disabled via bit ROMEN in register SYSCON. This bit is set during reset according to the level on pin EA, or may be altered via software. If enabled, the internal ROM area occupies the lower 32 KByte of either segment 0 or segment 1. This ROM mapping is controlled by bit ROMS1 in register SYSCON.

**Note:** The size of the internal ROM area is independent of the size of the actual implemented ROM. Also devices with less than 32 KByte of ROM or with no ROM at all will have this 32 KByte area occupied, if the ROM is enabled. Devices with larger ROMs provide the mapping option only for the ROM area.

Devices with a ROM size above 32 KByte expand the ROM area from the middle of segment 1, i.e. starting at address 01'8000H.

The internal ROM/Flash can be used for both code (instructions) and data (constants, tables, etc.) storage.

Code fetches are always made on even byte addresses. The highest possible code storage location in the internal ROM is either xx’xxFEH for single word instructions, or xx’xxFCH for double word instructions. The respective location must contain a branch instruction (unconditional), because sequential boundary crossing from internal ROM to external memory is not supported and causes erroneous results.

Any word and byte data read accesses may use the indirect or long 16-bit addressing modes. There is no short addressing mode for internal ROM operands. Any word data access is made to an even byte address. The highest possible word data storage location in the internal ROM is xx’xxFEH. For PEC data transfers the internal ROM can be accessed independent of the contents of the DPP registers via the PEC source and destination pointers.

The internal ROM is not provided for single bit storage, and therefore it is not bit addressable.

**Note:** The ‘x’ in the locations above depend on the available ROM/Flash memory and on the mapping.

The internal ROM may be enabled, disabled or mapped into segment 0 or segment 1 under software control. Chapter “System Programming” shows how to do this and reminds of the precautions that must be taken in order to prevent the system from crashing.
3.2 Internal RAM and SFR Area

The RAM/SFR area is located within data page 3 and provides access to 2 KByte of on-chip RAM (organized as $1K \times 16$) and to two 512 Byte blocks of Special Function Registers (SFRs). The internal RAM serves for several purposes:

- System Stack (programmable size)
- General Purpose Register Banks (GPRs)
- Source and destination pointers for the Peripheral Event Controller (PEC)
- Variable and other data storage, or
- Code storage.

Note: The upper 256 bytes of SFR area, ESFR area and internal RAM are bit-addressable (see shaded blocks in the figure above).
Code accesses are always made on even byte addresses. The highest possible code storage location in the internal RAM is either 00’FDFE_H for single word instructions or 00’FDFC_H for double word instructions. The respective location must contain a branch instruction (unconditional), because sequential boundary crossing from internal RAM to the SFR area is not supported and causes erroneous results.

Any word and byte data in the internal RAM can be accessed via indirect or long 16-bit addressing modes, if the selected DPP register points to data page 3. Any word data access is made on an even byte address. The highest possible word data storage location in the internal RAM is 00’FDFE_H. For PEC data transfers, the internal RAM can be accessed independent of the contents of the DPP registers via the PEC source and destination pointers.

The upper 256 Byte of the internal RAM (00’FD00_H through 00’FDFF_H) and the GPRs of the current bank are provided for single bit storage, and thus they are bit addressable.

**System Stack**

The system stack may be defined within the internal RAM. The size of the system stack is controlled by bitfield STKSZ in register SYSCON (see table below).

<table>
<thead>
<tr>
<th>&lt;STKSZ&gt;</th>
<th>Stack Size (Words)</th>
<th>Internal RAM Addresses (Words)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 B</td>
<td>256</td>
<td>00’FBFE_H...00’FA00_H         (Default after Reset)</td>
</tr>
<tr>
<td>0 0 1 B</td>
<td>128</td>
<td>00’FBFE_H...00’FB00_H</td>
</tr>
<tr>
<td>0 1 0 B</td>
<td>64</td>
<td>00’FBFE_H...00’FB80_H</td>
</tr>
<tr>
<td>0 1 1 B</td>
<td>32</td>
<td>00’FBFE_H...00’FBC0_H</td>
</tr>
<tr>
<td>1 0 0 B</td>
<td>512</td>
<td>00’FBFE_H...00’F800_H</td>
</tr>
<tr>
<td>1 0 1 B</td>
<td>---</td>
<td>Reserved. Do not use this combination.</td>
</tr>
<tr>
<td>1 1 0 B</td>
<td>---</td>
<td>Reserved. Do not use this combination.</td>
</tr>
<tr>
<td>1 1 1 B</td>
<td>1024</td>
<td>00’FDFE_H...00’F600_H         (Note: No circular stack)</td>
</tr>
</tbody>
</table>

For all system stack operations the on-chip RAM is accessed via the Stack Pointer (SP) register. The stack grows downward from higher towards lower RAM address locations. Only word accesses are supported to the system stack. A stack overflow (STKOV) and a stack underflow (STKUN) register are provided to control the lower and upper limits of the selected stack area. These two stack boundary registers can be used not only for protection against data destruction, but also allow to implement a circular stack with hardware supported system stack flushing and filling (except for the 2KByte stack option).

The technique of implementing this circular stack is described in chapter “System Programming”.

---

Semiconductor Group 3-5
General Purpose Registers

The General Purpose Registers (GPRs) use a block of 16 consecutive words within the internal RAM. The Context Pointer (CP) register determines the base address of the currently active register bank. This register bank may consist of up to 16 word GPRs (R0, R1, ..., R15) and/or up to 16 byte GPRs (RL0, RH0, ..., RL7, RH7). The sixteen byte GPRs are mapped onto the first eight word GPRs (see table below).

In contrast to the system stack, a register bank grows from lower towards higher address locations and occupies a maximum space of 32 bytes. The GPRs are accessed via short 2-, 4- or 8-bit addressing modes using the Context Pointer (CP) register as base address (independent of the current DPP register contents). Additionally, each bit in the currently active register bank can be accessed individually.

Mapping of General Purpose Registers to RAM Addresses

<table>
<thead>
<tr>
<th>Internal RAM Address</th>
<th>Byte Registers</th>
<th>Word Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;CP&gt; + 1E H</td>
<td>---</td>
<td>R15</td>
</tr>
<tr>
<td>&lt;CP&gt; + 1C H</td>
<td>---</td>
<td>R14</td>
</tr>
<tr>
<td>&lt;CP&gt; + 1A H</td>
<td>---</td>
<td>R13</td>
</tr>
<tr>
<td>&lt;CP&gt; + 18 H</td>
<td>---</td>
<td>R12</td>
</tr>
<tr>
<td>&lt;CP&gt; + 16 H</td>
<td>---</td>
<td>R11</td>
</tr>
<tr>
<td>&lt;CP&gt; + 14 H</td>
<td>---</td>
<td>R10</td>
</tr>
<tr>
<td>&lt;CP&gt; + 12 H</td>
<td>---</td>
<td>R9</td>
</tr>
<tr>
<td>&lt;CP&gt; + 10 H</td>
<td>---</td>
<td>R8</td>
</tr>
<tr>
<td>&lt;CP&gt; + 0E H</td>
<td>RH7</td>
<td>RL7</td>
</tr>
<tr>
<td>&lt;CP&gt; + 0C H</td>
<td>RH6</td>
<td>RL6</td>
</tr>
<tr>
<td>&lt;CP&gt; + 0A H</td>
<td>RH5</td>
<td>RL5</td>
</tr>
<tr>
<td>&lt;CP&gt; + 08 H</td>
<td>RH4</td>
<td>RL4</td>
</tr>
<tr>
<td>&lt;CP&gt; + 06 H</td>
<td>RH3</td>
<td>RL3</td>
</tr>
<tr>
<td>&lt;CP&gt; + 04 H</td>
<td>RH2</td>
<td>RL2</td>
</tr>
<tr>
<td>&lt;CP&gt; + 02 H</td>
<td>RH1</td>
<td>RL1</td>
</tr>
<tr>
<td>&lt;CP&gt; + 00 H</td>
<td>RH0</td>
<td>RL0</td>
</tr>
</tbody>
</table>

The C167 supports fast register bank (context) switching. Multiple register banks can physically exist within the internal RAM at the same time. Only the register bank selected by the Context Pointer register (CP) is active at a given time, however. Selecting a new active register bank is simply done by updating the CP register. A particular Switch Context (SCXT) instruction performs register bank switching and an automatic saving of the previous context. The number of implemented register banks (arbitrary sizes) is only limited by the size of the available internal RAM.

Details on using, switching and overlapping register banks are described in chapter “System Programming”.
PEC Source and Destination Pointers

The 16 word locations in the internal RAM from 00‘FCE0\text{H} to 00‘FCFE\text{H} (just below the bit-addressable section) are provided as source and destination address pointers for data transfers on the eight PEC channels. Each channel uses a pair of pointers stored in two subsequent word locations with the source pointer (SRCPx) on the lower and the destination pointer (DSTPx) on the higher word address (x = 7...0).

![Diagram of PEC Pointers]

**Figure 3-4**
Location of the PEC Pointers

Whenever a PEC data transfer is performed, the pair of source and destination pointers, which is selected by the specified PEC channel number, is accessed independent of the current DPP register contents and also the locations referred to by these pointers are accessed independent of the current DPP register contents. If a PEC channel is not used, the corresponding pointer locations area available and can be used for word or byte data storage.

For more details about the use of the source and destination pointers for PEC data transfers see section “Interrupt and Trap Functions”.

Semiconductor Group 3-7
Special Function Registers

The functions of the CPU, the bus interface, the IO ports and the on-chip peripherals of the C167 are controlled via a number of so-called Special Function Registers (SFRs). These SFRs are arranged within two areas of 512 Byte size each. The first register block, the SFR area, is located in the 512 Bytes above the internal RAM (00’FFFFH...00’FE00H), the second register block, the Extended SFR (ESFR) area, is located in the 512 Bytes below the internal RAM (00’F1FFH...00’F000H).

Special function registers can be addressed via indirect and long 16-bit addressing modes. Using an 8-bit offset together with an implicit base address allows to address word SFRs and their respective low bytes. However, this does not work for the respective high bytes!

Note: Writing to any byte of an SFR causes the non-addressed complementary byte to be cleared!

The upper half of each register block is bit-addressable, so the respective control/status bits can directly be modified or checked using bit addressing.

When accessing registers in the ESFR area using 8-bit addresses or direct bit addressing, an Extend Register (EXTR) instruction is required before, to switch the short addressing mechanism from the standard SFR area to the Extended SFR area. This is not required for 16-bit and indirect addresses. The GPRs R15...R0 are duplicated, ie. they are accessible within both register blocks via short 2-, 4- or 8-bit addresses without switching.

Example:

```
EXTR  #4 ;Switch to ESFR area for the next 4 instructions
MOV   ODP2, #data16 ;ODP2 uses 8-bit reg addressing
BFLDL DP6, #mask, #data8 ;Bit addressing for bit fields
BSET  DP1H.7 ;Bit addressing for single bits
MOV   T8REL, R1 ;T8REL uses 16-bit address, R1 is duplicated...
               ;...and also accessible via the ESFR mode
               ;(EXTR is not required for this access)

;------ ;------------------- ;The scope of the EXTR #4 instruction ends here!

MOV   T8REL, R1 ;T8REL uses 16-bit address, R1 is duplicated...
               ;...and does not require switching
```

In order to minimize the use of the EXTR instructions the ESFR area mostly holds registers which are mainly required for initialization and mode selection. Registers that need to be accessed frequently are allocated to the standard SFR area, wherever possible.

Note: The tools are equipped to monitor accesses to the ESFR area and will automatically insert EXTR instructions, or issue a warning in case of missing or excessive EXTR instructions.
3.3 The On-Chip XRAM

The XRAM area is located within data page 3 and provides access to 2 KByte of on-chip RAM (organized as 1K*16). As the XRAM is connected to the internal XBUS it is accessed like external memory, however, no external bus cycles are executed for these accesses. XRAM accesses are globally enabled or disabled via bit XPEN in register SYSCON. This bit is cleared after reset and may be set via software during the initialization to allow accesses to the on-chip XRAM. When the XRAM is disabled (default after reset) all accesses to the XRAM area are mapped to external locations. The XRAM may be used for both code (instructions) and data (variables, user stack, tables, etc.) storage.

Code fetches are always made on even byte addresses. The highest possible code storage location in the XRAM is either 00'E7FEH for single word instructions, or 00'E7FCH for double word instructions. The respective location must contain a branch instruction (unconditional), because sequential boundary crossing from XRAM to external memory is not supported and causes erroneous results.

Any word and byte data read accesses may use the indirect or long 16-bit addressing modes. There is no short addressing mode for XRAM operands. Any word data access is made to an even byte address. The highest possible word data storage location in the XRAM is 00'E7FEH. For PEC data transfers the XRAM can be accessed independent of the contents of the DPP registers via the PEC source and destination pointers.

Note: As the XRAM appears like external memory it cannot be used for the C167’s system stack or register banks. The XRAM is not provided for single bit storage and therefore is not bit addressable.

The on-chip XRAM is accessed without any waitstates using 16-bit demultiplexed bus cycles which take 100 ns (@ 20 MHz fCPU). Even if the XRAM is used like external memory it does not occupy BUSCONx/ADDRSELx registers but rather is selected via additional dedicated XBCON/XADRS registers. These registers are mask-programmed and are not user accessible. With these registers the address area 00’E000H to 00’E7FFH is reserved for XRAM accesses.

XRAM Access via External Masters

When bit XPER-SHARE in register SYSCON is set the on-chip XRAM of the C167 can be accessed by an external master during hold mode via the C167’s bus interface. These external accesses must use the same configuration as internally programmed, ie. demultiplexed bus, 100 ns minimum access cycle time. No waitstates are required.

Note: The configuration in register SYSCON cannot be changed after the execution of the EINIT instruction.
Figure 3-5
On-chip XRAM Area

Note: The address area 00'800H to 00'EFFH is mapped to external memory but should be reserved for reasons of upward compatibility.
3.4 External Memory Space

The C167 is capable of using an address space of up to 16 MByte. Only parts of this address space are occupied by internal memory areas. All addresses which are not used for on-chip memory (ROM or RAM) or for registers may reference external memory locations. This external memory is accessed via the C167’s external bus interface.

Four memory bank sizes are supported:

- Non-segmented mode: 64 KByte with A15...A0 on PORT0 or PORT1
- 2-bit segmented mode: 256 KByte with A17...A16 on Port 4 and A15...A0 on PORT0 or PORT1
- 4-bit segmented mode: 1 MByte with A19...A16 on Port 4 and A15...A0 on PORT0 or PORT1
- 8-bit segmented mode: 16 MByte with A23...A16 on Port 4 and A15...A0 on PORT0 or PORT1

Each bank can be directly addressed via the address bus, while the programmable chip select signals can be used to select various memory banks.

The C167 also supports four different bus types:

- Multiplexed 16-bit Bus with address and data on PORT0 (Default after Reset)
- Multiplexed 8-bit Bus with address and data on PORT0/P0L
- Demultiplexed 16-bit Bus with address on PORT1 and data on PORT0
- Demultiplexed 8-bit Bus with address on PORT1 and data on P0L

Memory model and bus mode are selected during reset by pin EA and PORT0 pins. For further details about the external bus configuration and control please refer to chapter "The External Bus Interface".

External word and byte data can only be accessed via indirect or long 16-bit addressing modes using one of the four DPP registers. There is no short addressing mode for external operands. Any word data access is made to an even byte address.

For PEC data transfers the external memory in segment 0 can be accessed independent of the contents of the DPP registers via the PEC source and destination pointers.

The external memory is not provided for single bit storage and therefore it is not bit addressable.
3.5 Crossing Memory Boundaries

The address space of the C167 is implicitly divided into equally sized blocks of different granularity and into logical memory areas. Crossing the boundaries between these blocks (code or data) or areas requires special attention to ensure that the controller executes the desired operations.

**Memory Areas** are partitions of the address space that represent different kinds of memory (if provided at all). These memory areas are the internal RAM/SFR area, the internal ROM (if available), the on-chip X-Peripherals (if integrated) and the external memory.

Accessing subsequent data locations that belong to different memory areas is no problem. However, when executing code, the different memory areas must be switched explicitly via branch instructions. Sequential boundary crossing is not supported and leads to erroneous results.

**Note:** Changing from the external memory area to the internal RAM/SFR area takes place within segment 0.

**Segments** are contiguous blocks of 64 KByte each. They are referenced via the code segment pointer CSP for code fetches and via an explicit segment number for data accesses overriding the standard DPP scheme.

During code fetching segments are not changed automatically, but rather must be switched explicitly. The instructions JMPS, CALLS and RETS will do this.

In larger sequential programs make sure that the highest used code location of a segment contains an unconditional branch instruction to the respective following segment, to prevent the prefetcher from trying to leave the current segment.

**Data Pages** are contiguous blocks of 16 KByte each. They are referenced via the data page pointers DPP3...0 and via an explicit data page number for data accesses overriding the standard DPP scheme. Each DPP register can select one of the possible 1024 data pages. The DPP register that is used for the current access is selected via the two upper bits of the 16-bit data address.

Subsequent 16-bit data addresses that cross the 16 KByte data page boundaries therefore will use different data page pointers, while the physical locations need not be subsequent within memory.