# Segmentation

The processor contains sixteen segment registers. The upper nibble of an address (bits 60 to 63) identifies which segment register to use during address formation.

* If segmentation is not desired then segmentation can effectively be ignored by setting all the segment registers to zero. The processor can also be built without segmentation by commenting out the ‘SEGMENTATION’ definition.

# Software Support

Segment registers may only be transferred to or from one of the general purpose registers. There are four instructions for loading and storing the segment registers. Mtseg (move to segment), mtsegi (move to segment indirect),mfseg (move from segment, and mfsegi (move from segment indirect).

The mtseg and mfseg instructions transfer values between a general purpose register and a segment register directly. The mtsegi and mfsegi transfer values between a general purpose register and a segment register identified indirectly by another general purpose register. For the mtsegi and mfsegi instruction the upper nibble of the value in a register identifies which segment register to use in the transfer.

# Address Formation:

Non-segmented address bits 0 to 11 pass through the segmentation module unchanged. Address bits 59 to 12 are added to the contents of the segment register to form the final segmented address. Address bits 60 to 63 identify the segment register to use. Note that there is no shift associated with the segment addition. Future implementations of the processor may include additional low order address bits in the segment register in order to allow a finer grain for memory page / paragraoh size.

|  |  |
| --- | --- |
| Address[59:12] | Address[11:0] |
| + | + |
| Segment register value[63:12] | 00012 |
| = | |
| Segmented address[63:0] | |

## Selecting a segment register

The upper nibble of an address (bits 60 to 63) identifies which segment register to use.