Circular Buffer Implementation

As an example of how to set up a circular buffer, consider this procedure for a circular buffer of words in data memory:

1) Initialize the appropriate buffer size register (BK03, BK47, or BKC). For example, for a buffer of size 8, load the BK register with 8.
2) Initialize the appropriate configuration bit in ST2_55 to choose circular modification for the selected pointer.
3) Initialize the appropriate extended register (XARn or XCDP) to select a main data page (in the 7 most significant bits). For example, if auxiliary register 3 (AR3) is the circular pointer, load extended auxiliary register 3 (XAR3). If CDP is the circular pointer, load XCDP.
4) Initialize the appropriate buffer start address register (BSA01, BSA23, BSA45, BSA67, or BSAC). The main data page, in XARn(22-16) or XCDP(22-16), concatenated with the content of the BSA register defines the 23-bit start address of the buffer.
5) Load the selected pointer, ARn or CDP, with a value from **0 to (buffer size - 1)**. For example, if you are using AR1 and the buffer size is 8, load AR1 with a value less than or equal to 7.

A circular buffer of size R must start on an N-bit boundary, where N is the smallest integer that satisfies the relationship, 2N > R. For example, for a buffer size R = 8, N is 4. In this case, the top of the circular buffer is the address generated when the 4 LSBs of the pointer (ARn or CDP) are 0s. When the address incrementing leads beyond the buffer, the 4 LSBs of the pointer are forced to 0s.

If you are using indirect addressing operands with offsets, make sure that the absolute value of each offset is less than or equal to (buffer size - 1). Likewise, if the circular pointer is to be incremented or decremented by a programmed amount (supplied by a constant or by T0, AR0, or T1), make sure the absolute value of that amount is less than or equal to (buffer size - 1).

After the initialization, you have a 23-bit address of the following form:

ARnH:(BSAxx + ARn),
or
CDPH:(BSAC + CDP)

Increments and decrements are made to the 16-bit pointer (ARn or CDP) only. You cannot address data across main data pages without changing the value in the corresponding extension register (ARnH or CDPH).

Note:

> Although an increment past FFFFh or a decrement past 0000h causes the pointer value to wrap around, do not make use of this behavior; it is not supported. Also, the BSAxx or BSAC addition must not increment the address beyond FFFFh.

The following code example demonstrates initializing and then accessing a circular buffer.

Initializing and Accessing a C55x Circular Buffer

```
MOV #3, BK03              ; Circular buffer size is 3 words
BSET AR1LC                ; AR1 is configured to be modified circularly
AMOV #010000h, XAR1       ; Circular buffer is in main data page 01
MOV #0A02h, BSA01         ; Circular buffer start address is 010A02h
MOV #0000h, AR1           ; Index (in AR1) is 0000h
MOV *AR1+, AC0            ; AC0 loaded from 010A02h + (AR1) = 010A02h,
                          ; and then AR1 = 0001h
MOV *AR1+, AC0            ; AC0 loaded from 010A02h + (AR1) = 010A03h,
                          ; and then AR1 = 0002h
MOV *AR1+, AC0            ; AC0 loaded from 010A02h + (AR1) = 010A04h,
                          ; and then AR1 = 0000h
MOV *AR1+, AC0            ; AC0 loaded from 010A02h + (AR1) = 010A02h,
                          ; and then AR1 = 0001h
```