```
        CLR     A
        CLR     B
        MOVE    B1,Y:<TEST1
        MOVE    Y:<NROWS,A1
        DEC     A                        ; NROWS - 1
        DO      A1,FRAME                 ; clock-and-read the remaining (1023) rows per half
        MOVE    Y:<TEST1,A1
        MOVE    X:<ONE,X1                ; Increment row counter
        ADD     X1,A            ; Increment row counter
        MOVE    A1,Y:<TEST1     ; save row counter

        ;; Set HCLK low at the end of each row before the next row is clocked
        ;; This is what is shown in the Rockwell timing diagram.
        MOVE    #HCLOCK_LOW,R0
        JSR     <CLOCK

        JSET    #RST_MODE,X:STATUS,RRR_RD1
        MOVE    #CLOCK_ROW,R0            ;
        JSR     <CLOCK
        JMP     <RRR_RD2
        ; **** this should only happen on the first readout. Need to fix this for row-by-row (bad
 comment-MWR)*******
RRR_RD1 MOVE    #CLOCK_ROW_RESET,R0      ; if row-by-row reset
        JSR     <CLOCK
RRR_RD2 NOP

        MOVE    #READ_ON,R0             ; Turn Read ON
        JSR     <CLOCK

        ; for 1RG, need 2 HCLK pulses before the first real pixel in each row
        ; This affects the relative start col of each row the same in all reads
        ; It looks like 2 HCLKs is correct.
        DO      #2,SKIP_1       ; comment
        MOVE    #CLOCK_ONLY,R0  ; comment
        JSR     <CLOCK          ; comment
                ;;
        NOP                     ; r
SKIP_1


; use this block for COL skipping  (this is for burst-mode subarray)
        MOVE    Y:<STARTCOL,A
        MOVE    X:ONE,X0
        CMP     X0,A                            ;   if # of cols to skip is zero, then don't
        JEQ     <NOCOLSKIP
        DEC     A
        DO      A,COLSKIP
        MOVE    #CLOCK_ONLY_FAST,R0
        JSR     <CLOCK
        NOP
COLSKIP
        NOP
NOCOLSKIP

        MOVE    Y:<TEMP,X0               ; total cols/2
        DO      X0,L_READ_LAST_TWO_PIXELS ; clock-and-read (512) pix per row, 1 at a time
        JSET    #BO_MODE,X:STATUS,RR_BO2      ; if BO mode selected, jump to the BO method
        MOVE    Y:<CLK_COL_AND_READ,R0
        JSR     <CLOCK
        JMP     <OVER_BO2       ; jump over BO mode

; ****************************************************************************
; this is BO mode
RR_BO2  MOVE    #CLOCK_ONLY,R0     ; clock to pixel
        JSR     <CLOCK
        NOP
        MOVE    #PULSE_RESET,R0          ; strobe the reset on this row
        JSR     <CLOCK
        NOP


        ; put a settling time delay here (10 us)
```

```
        DO      #200,SET_DLY4               ; Each step of this loop is 1 us
        NOP
SET_DLY4

        JCLR    #CDS_MODE,X:STATUS,BO_RD_END_4
        DO      Y:<NFS,BO_RD_END_4                          ; Fowler sampling

        ; first BO read(s)
        MOVE    #A2D_ONLY,R0
        JSR     <CLOCK

        NOP
BO_RD_END_4

; Implement up-the-ramp readout
        DO      Y:<IUTR,BO_L_UP_THE_RAMP_4

        ; delay loop goes here for exptime in BO mode
        ; this is the number to delay by -  X:<EXP_TIM
        DO      Y0,EXP_DLY4               ; Each step of this loop is 1 us
        NOP
EXP_DLY4

; Sample the signal level NFS times
        DO      Y:<NFS,BO_RD_EXP_4                          ; Fowler sampling

        ; second BO read
        MOVE    #A2D_ONLY,R0
        JSR     <CLOCK

        NOP
BO_RD_EXP_4
        NOP
BO_L_UP_THE_RAMP_4
        NOP
; *****************************************************************************
OVER_BO2
        NOP
L_READ_LAST_TWO_PIXELS
```