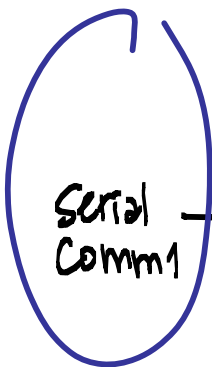


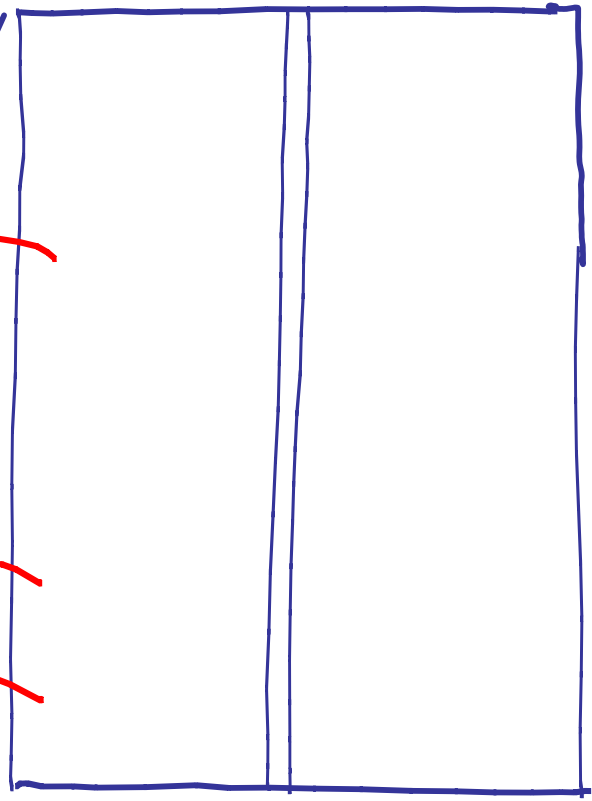
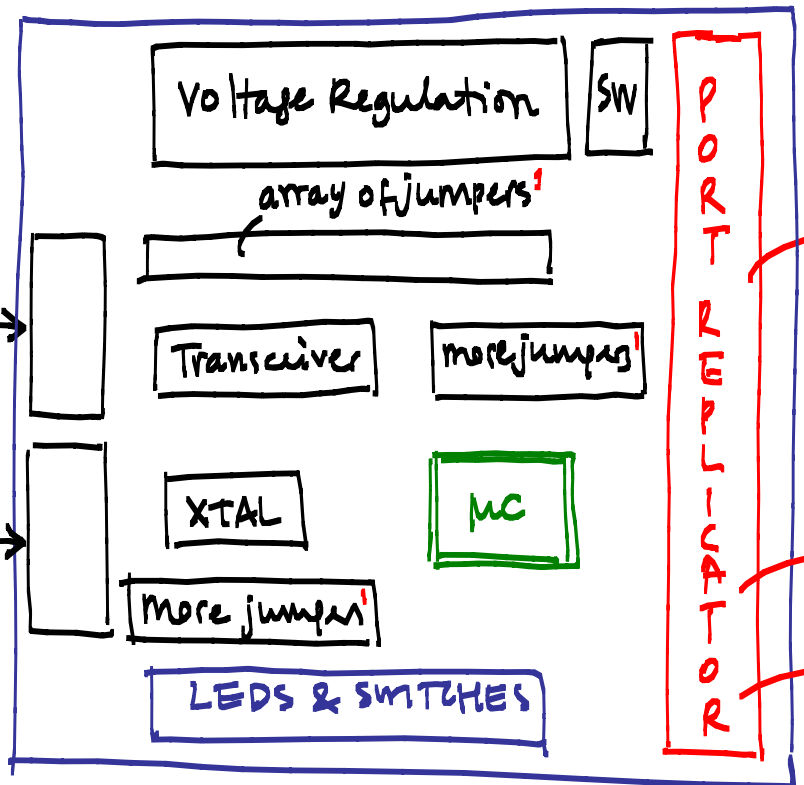
# THE DEMONSTRATION/REFERENCE BOARD

EVALUATION BOARD

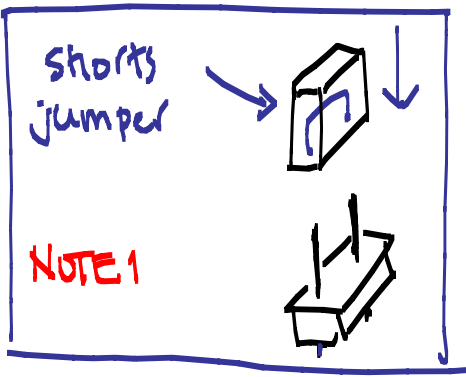
BREADBOARD



Serial Comm2

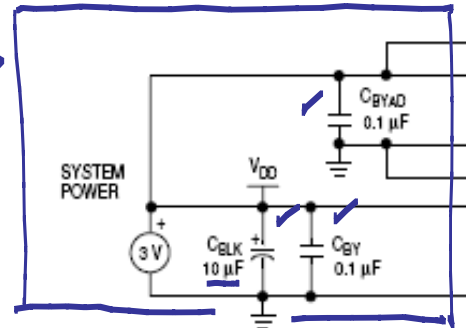


Wires that connect to external Hardware

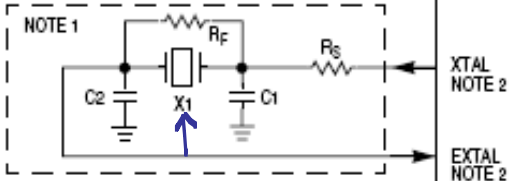


# BLOCK DIAGRAM OF THE MICROCONTROLLER I/O PINS

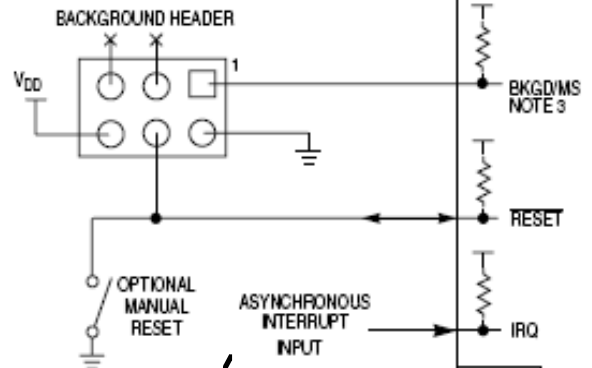
Power cleaning circuitry



XTAL



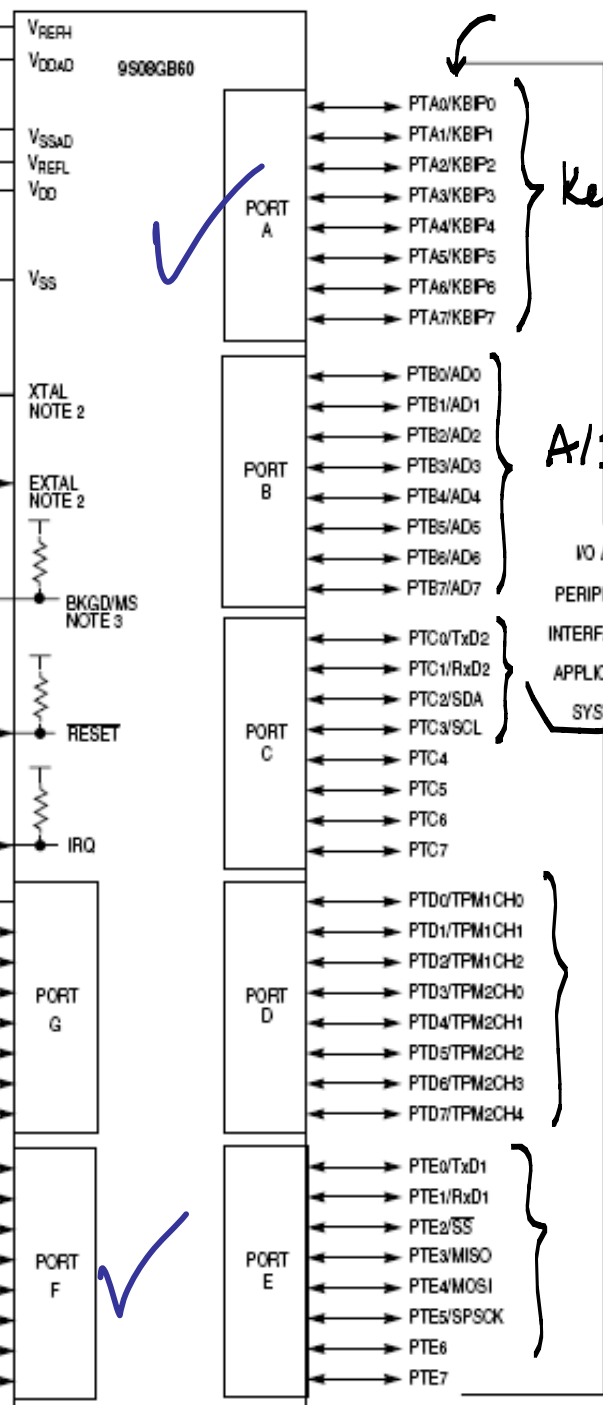
Operation mode configuration



XTAL

only I/O

- NOTES:
1. Not required if using the internal oscillator option.
  2. These are the same pins as PTG1 and PTG2.
  3. BKGD/MS is the same pin as PTG0.



Key pad functionality

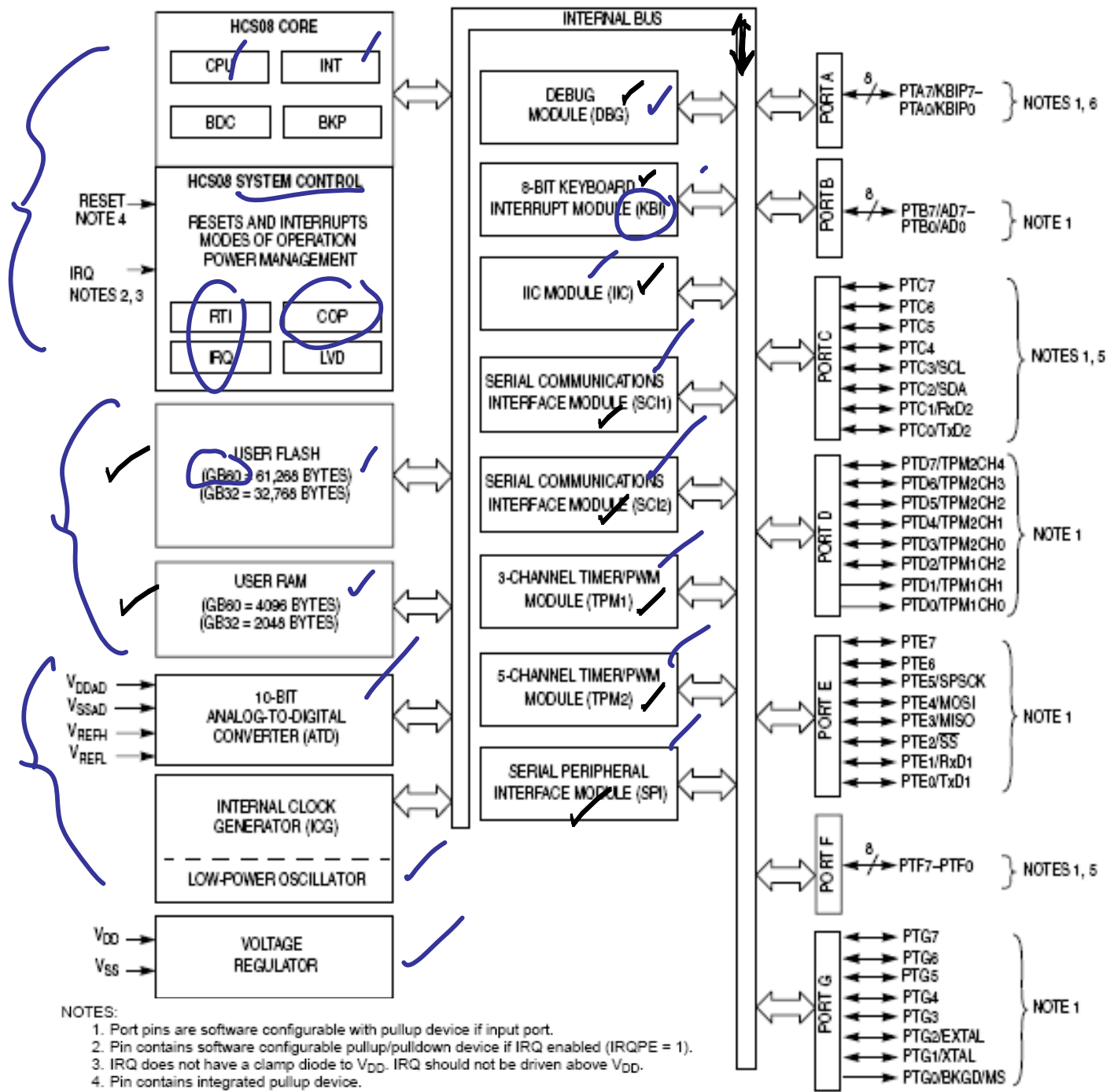
A/D pins

IO AND PERIPHERAL INTERFACE TO APPLICATION SYSTEM  
TX/RX serial

Timer functionality

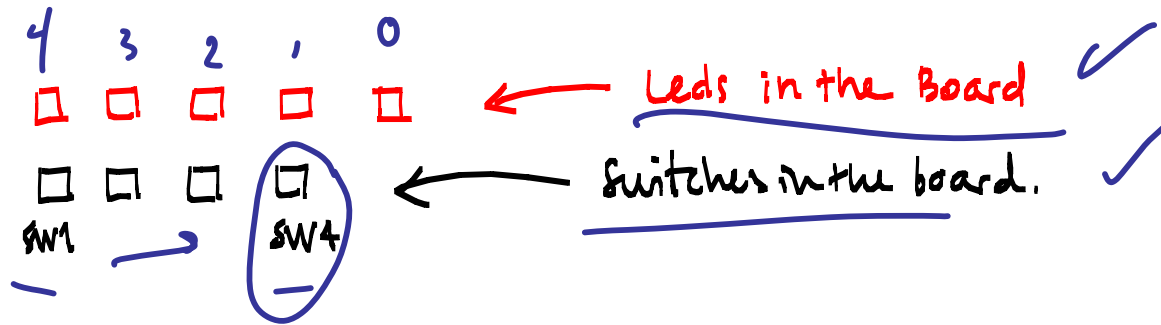
Tx/Rx

# DETAIL ON MC MODULES



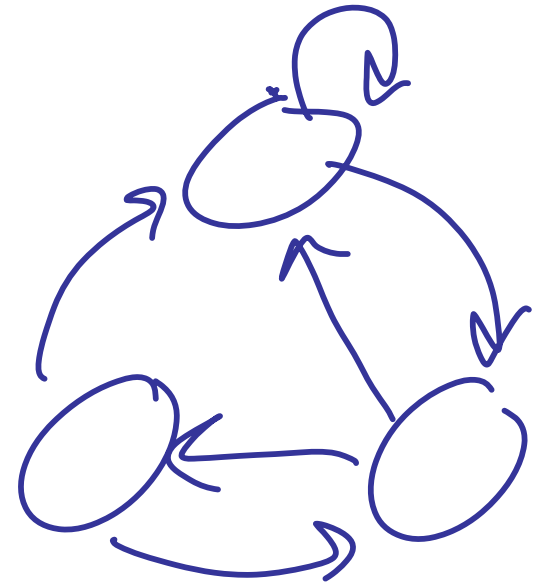
- NOTES:
1. Port pins are software configurable with pullup device if input port.
  2. Pin contains software configurable pullup/pulldown device if IRQ enabled (IRQPE = 1).
  3. IRQ does not have a clamp diode to V<sub>DD</sub>. IRQ should not be driven above V<sub>DD</sub>.
  4. Pin contains integrated pullup device.
  5. High current drive
  6. Pins PTA[7:4] contain software configurable pullup/pulldown device.

# PROJECT 0 DEFINITION



## Project description/requirements

- Mode 1 - Reset/Initial
  - All LED's are OFF.
  - SW4 is pressed - enter Mode 2.
  - Ignore any button presses other than SW4.
- Mode 2 - Start
  - LED4 is ON.
  - LED1, LED2, LED3 count up in Binary, about once every second.
  - SW3 is pressed - enter Mode 3.
  - Ignore any button presses other than SW3.
- Mode 3 - Stop
  - LED4 is OFF.
  - LED1, LED2, LED3 remain showing last value.
  - SW4 is pressed - enter Mode 2.
  - SW3 is pressed - remain in Mode 3
  - Any other button presses (SW1, SW2) - enter Mode 1



# DESIGN METHODOLOGY

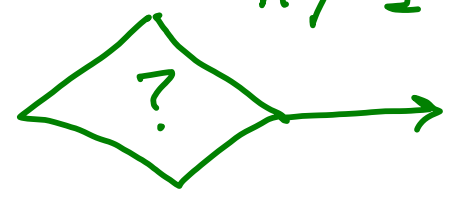
Phase	Description	Outcome	
∅	Intro to the problem	UNDERSTANDING.	✓
1	Hardware discussion Which components How to connect them Anticipate for debugging	HWa Proven HW.	N/A ✓
2	Software discussion Flow diagram/architecture variable definition CODING	HWb	?
3	Implementation / Lab demo	Working solution	✓
4	Documentation	Design Notebook.	✓

# PROJECT Ø - FLOW DIAGRAM

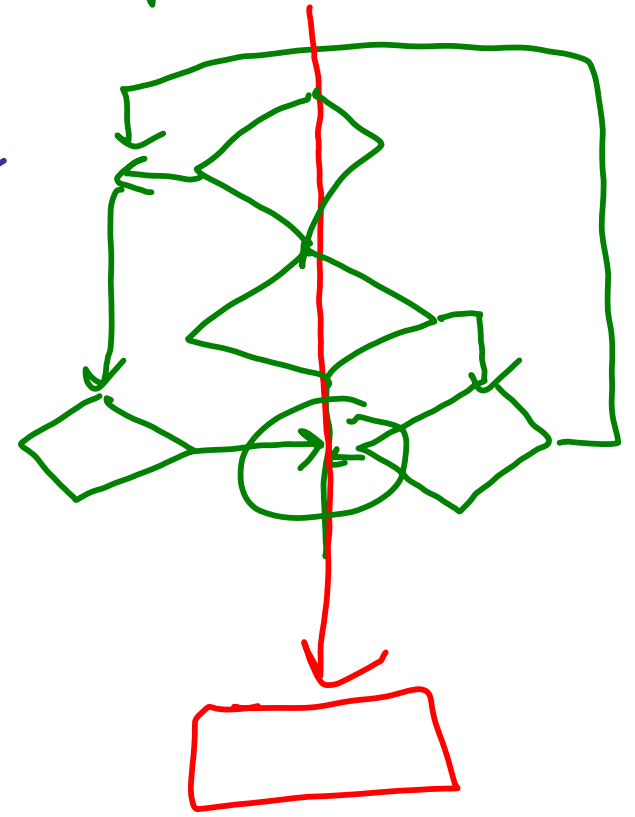
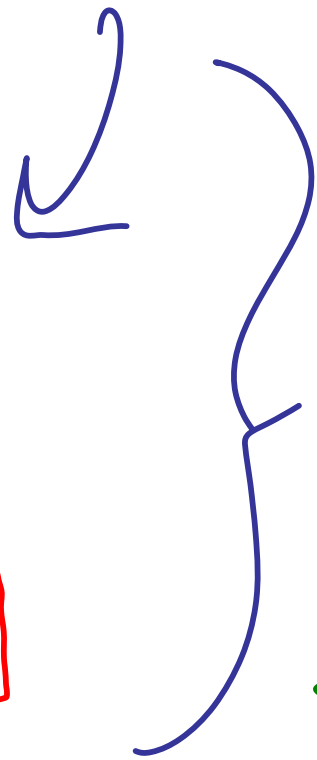
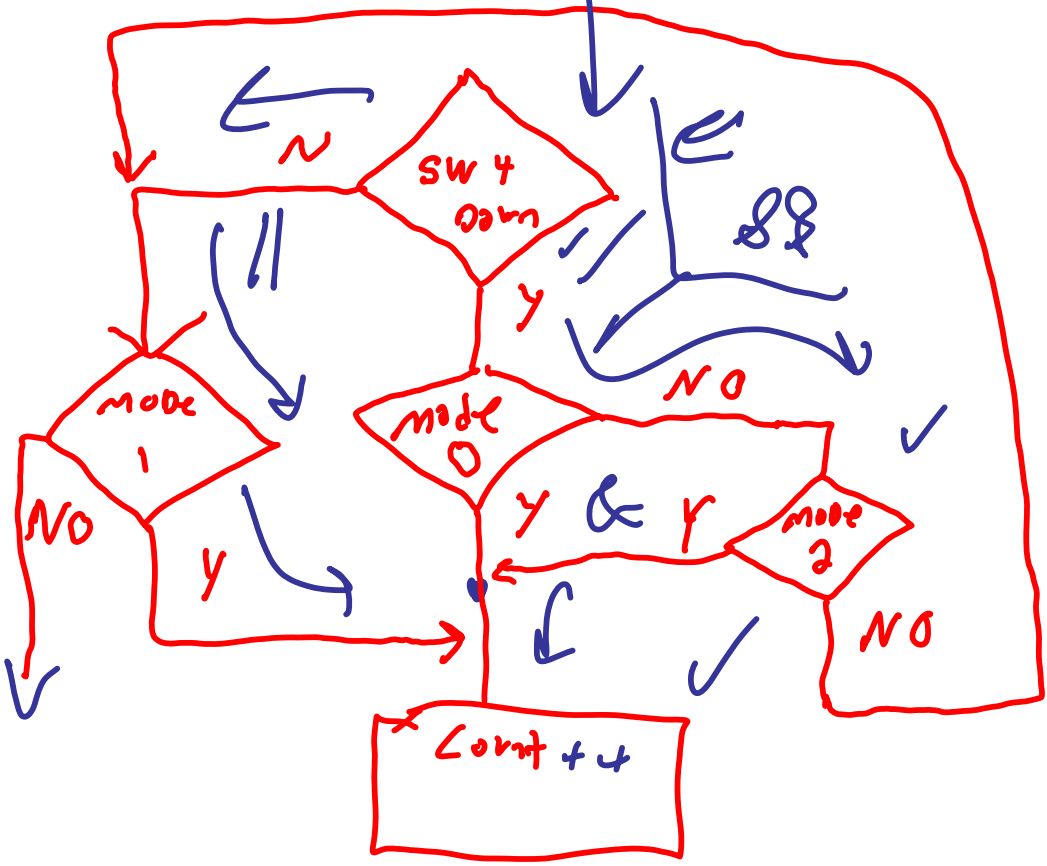
```

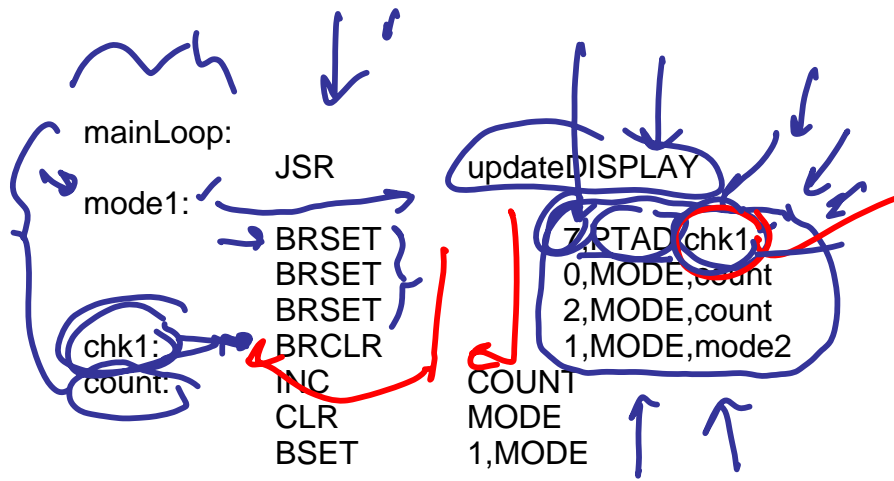
if ((SW4==DOWN && (mode==0 || mode==2)) || mode==1)
{
  mode=1;
  LED4=ON;
  displayLEDs();
  count++;
}
  
```

Decision Box only 1 out



Design flow to limit Branches



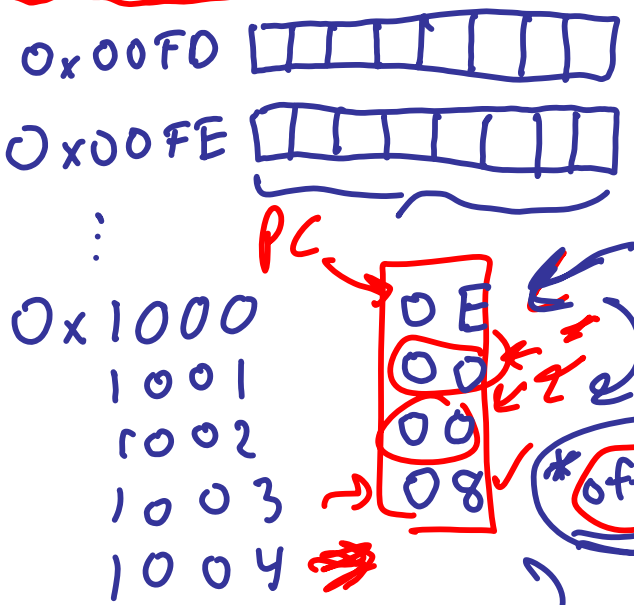


```

; Check if SW4 is pressed
; Check if mode0
; Check if mode2
; Check if mode1
; count++
; set mode=1

```

Label: Inst Operand(s) ; Comment



```

→ PTAD == 0x0000 / Def.
FLASH / Mode1 == 0x1000 ← ADDR
Variable { MODE == 0x00FD } RAM
           { COUNT == 0x00FE } RAM
FLASH → CHK1 == 0x100C

```

ADDRESSES

ADDR: DATA  
86

# BRSET *n*

## Branch if Bit *n* in Memory Set

# BRSET *n*

### Operation

If bit *n* of M = 1,  $PC \leftarrow (PC) + \$0003 + rel$

### Description

Tests bit *n* (*n* = 7, 6, 5, ... 0) of location M and branches if the bit is set. M can be any RAM or I/O register address in the \$0000 to \$00FF area of memory because direct addressing mode is used to specify the address of the operand.

The C bit is set to the state of the tested bit. When used with an appropriate rotate instruction, BRSET *n* provides an easy method for performing serial-to-parallel conversions.

### Condition Codes and Boolean Formulae

V	H	I	N	Z	C
—	1	1	—	—	!

C: Set if  $M_n = 1$ ; cleared otherwise

### Source Forms, Addressing Modes, Machine Code, Cycles, and Access Details

Source Form	Addr. Mode	Machine Code			HCS08 Cycles	Access Detail
		Opcode	Operand(s)			
BRSET 0,opr8a,rel	DIR (b0)	00	dd	rr	5	rpppp
BRSET 1,opr8a,rel	DIR (b1)	02	dd	rr	5	rpppp
BRSET 2,opr8a,rel	DIR (b2)	04	dd	rr	5	rpppp
BRSET 3,opr8a,rel	DIR (b3)	06	dd	rr	5	rpppp
BRSET 4,opr8a,rel	DIR (b4)	08	dd	rr	5	rpppp
BRSET 5,opr8a,rel	DIR (b5)	0A	dd	rr	5	rpppp
BRSET 6,opr8a,rel	DIR (b6)	0C	dd	rr	5	rpppp
BRSET 7,opr8a,rel	DIR (b7)	0E	dd	rr	5	rpppp

\* Goto Demo board Ref.

