Branch, Call & Time-delay loop

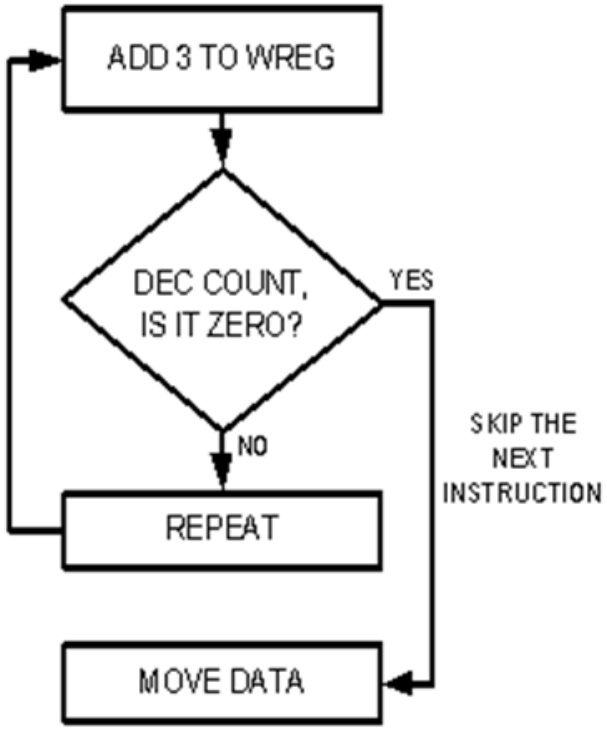
Hsiao-Lung Chan
Dept Electrical Engineering
Chang Gung University, Taiwan
chanhl@mail.cgu.edu.tw

Objectives

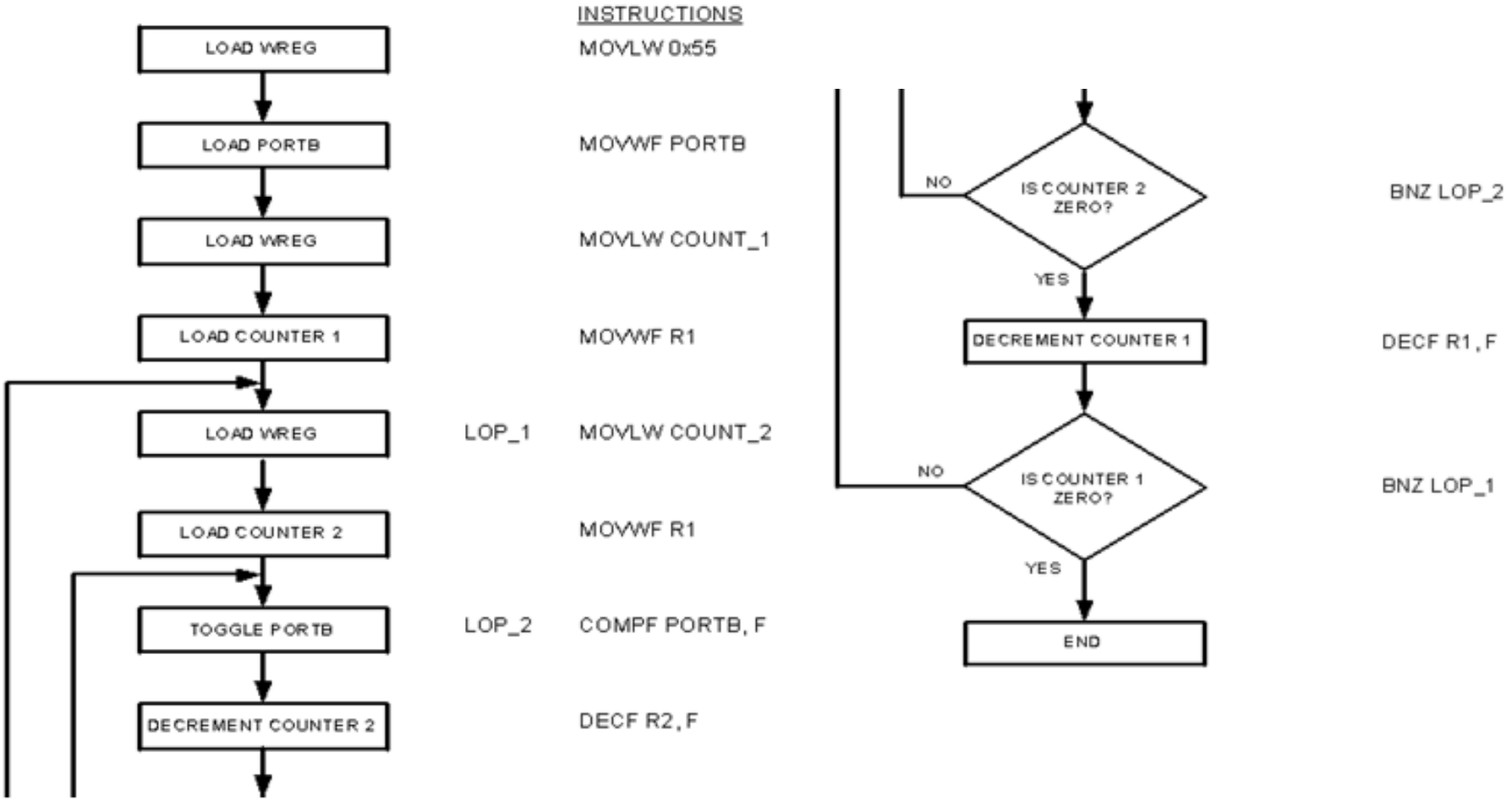
- Unconditional branch (GOTO, BRA)
  Conditional branch (BC, BNC, BZ, BNZ, BN, BNN, BOV, BNOV)
- Create loops (single, nested)
- Call a function (CALL, RCALL)
- Generate a time delay using loops

Looping

1. Single loop (Examples 3-1 & 3-2)
2. Loop inside a loop (Example 3-4)
3. Looping 100,000 times



Loop inside a loop



Branch

- Unconditional branch instructions
  - GOTO (GOTO is a long jump)
  - BRA (BRA is a short jump)
- Conditional branch instructions
  - All are short jumps

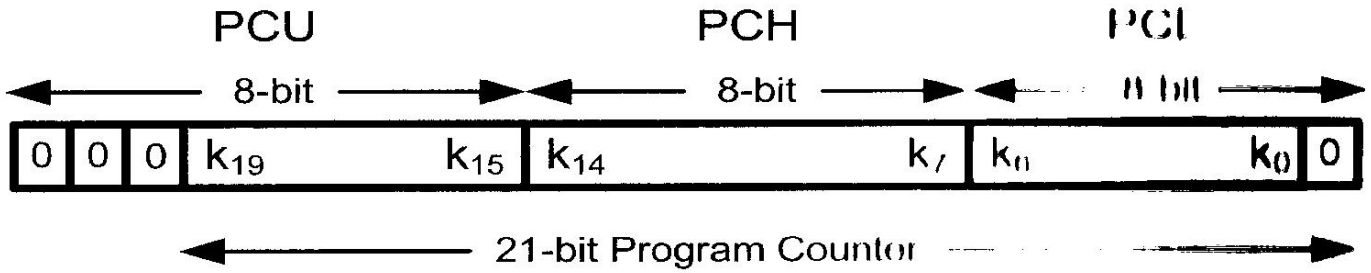
GOTO instruction

- 32-bit instruction

1110	1111	k_7kkk	$kkkk_0$
1111	$k_{19}kkk$	kkkk	$kkkk_8$

$$0 \leq k \leq FFFFF$$

- PIC16 has 2M of ROM space
- The LSB is 0, making sure even address

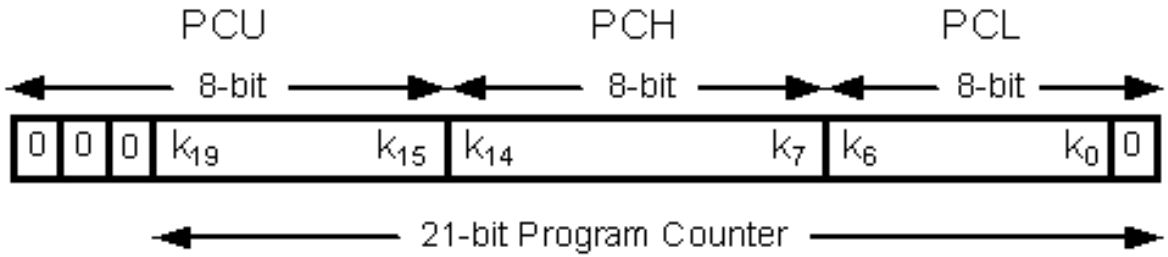


Goto vs. bra

- **goto k**
  Jump to absolute address

1110	1111	k_7kkk	$kkkk_0$
1111	$k_{19}kkk$	kkkk	$kkkk_8$

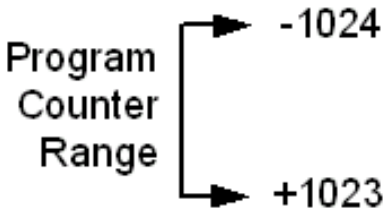
$0 \leq k \leq FFFFF$



- **bra**
  Jump to relative address
  (PC+n)

1101	0nnn	nnnn	nnnn

$-1024 \leq n \leq 1023$



Branch instruction using flag bits

Instruction	Action
BC	Branch if C = 1
BNC	Branch if C ≠ 1
BZ	Branch if Z = 1
BNZ	Branch if Z ≠ 1
BN	Branch if N = 1
BN N	Branch if N ≠ 1
BOV	Branch if OV = 1
BNOV	Branch if OV ≠ 1

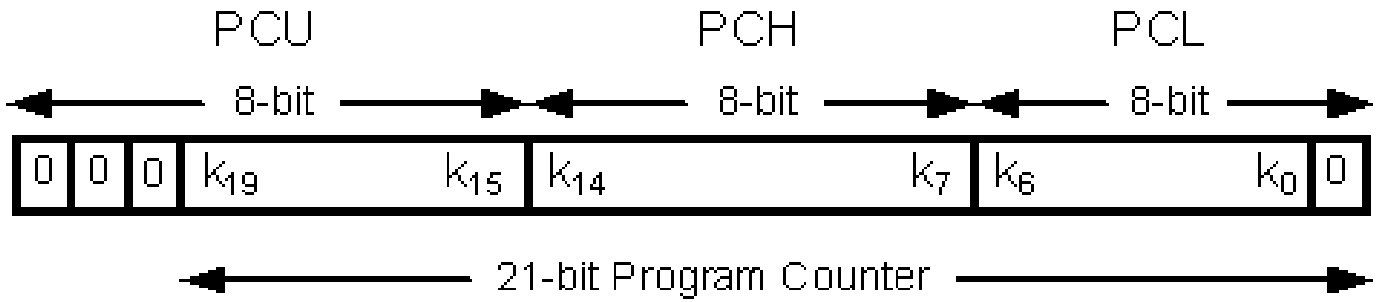
CALL instructions

- Jump and return (the role of the stack)
- Call a subroutine from the main program (Example 3-9)
- Call many subroutines from the main program (Example 3-11)
- Relative call (short jump and return) (Example 3-12)

Call instruction

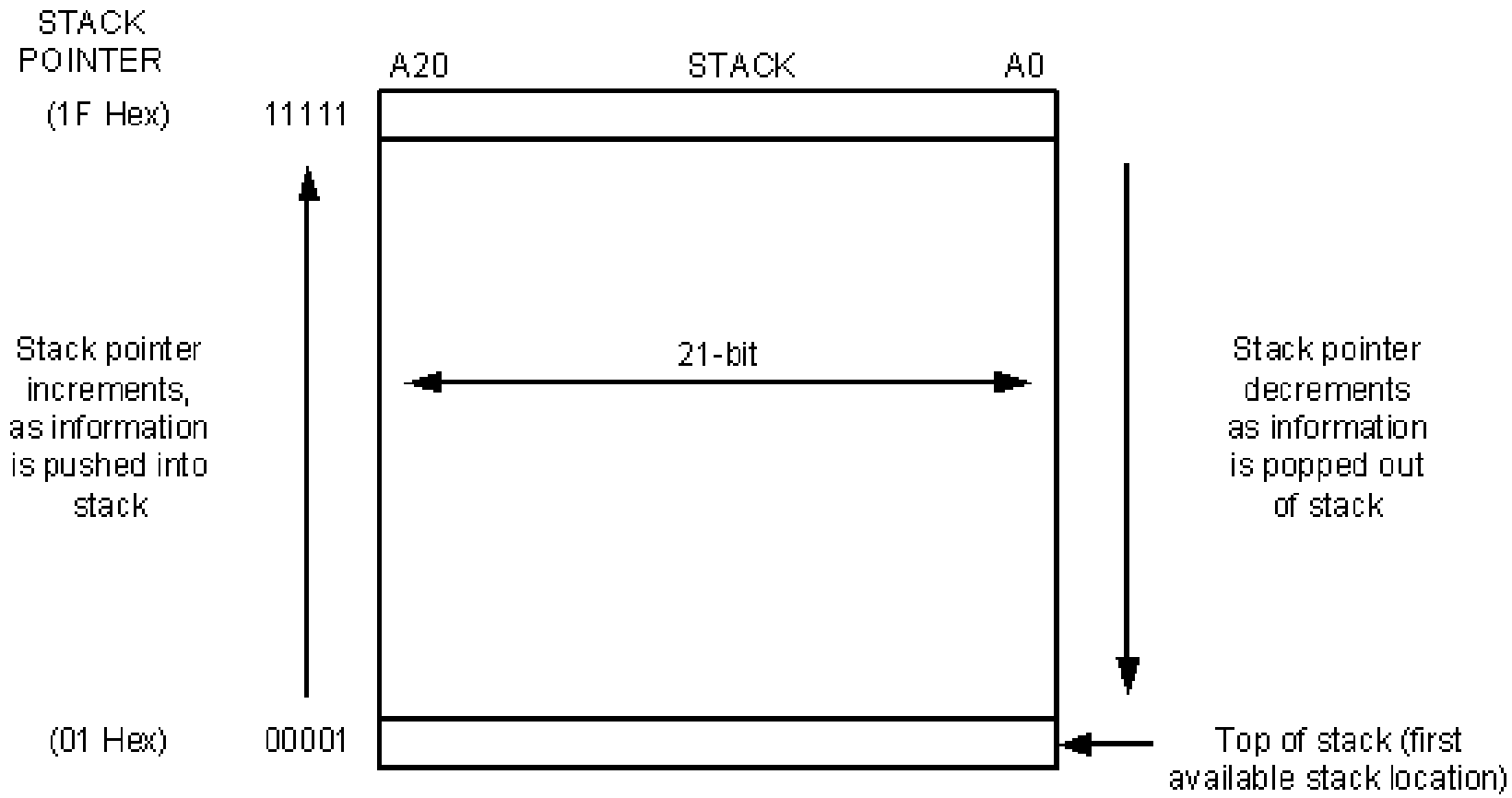
1110	1110	k_7kkk	$kkkk_0$
1111	$k_{19}kkk$	kkkk	$kkkk_8$

$0 \leq k \leq FFFFF$



Stack and stack pointer

- Store critical information temporarily



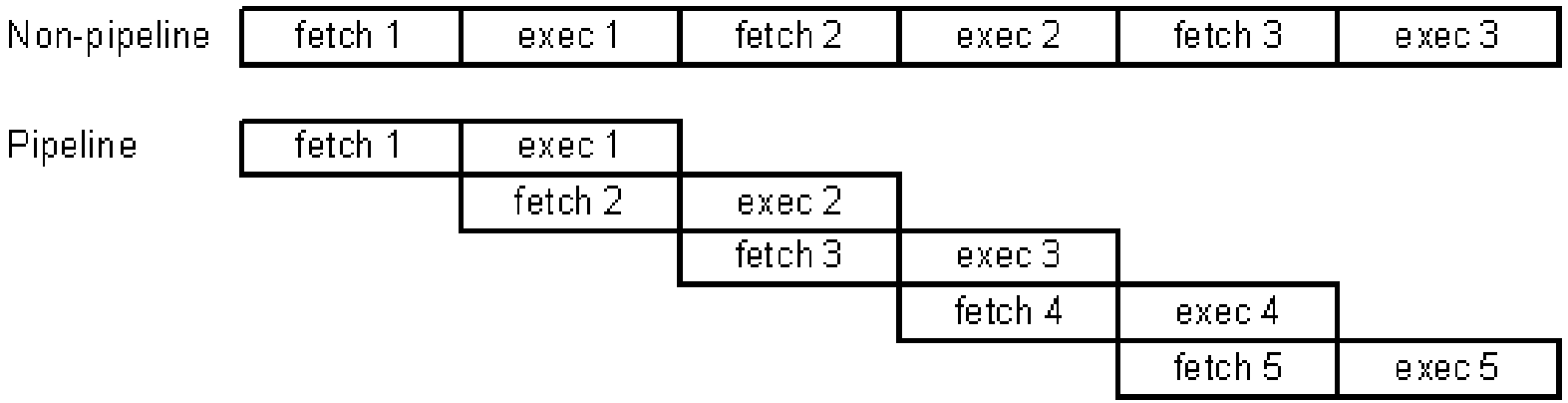
Generate a time delay

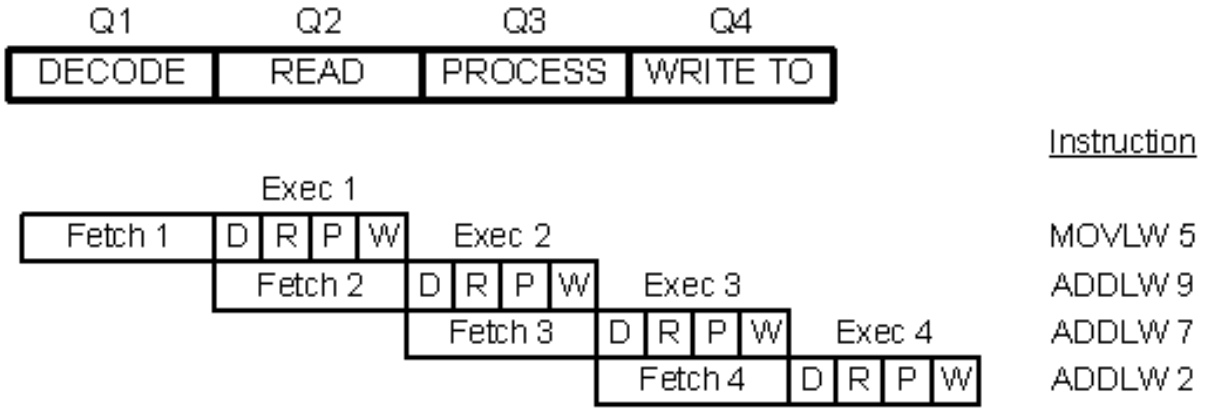
- Instruction cycle
- Delay calculation (Example 3-18)
- Large delay using a nested loop (Example 3-20)

Instruction cycle

- In Pic18, one instruction cycle consists of 4 oscillatory cycles
- Example: Find the period of instruction cycle if crystal frequency is 4 MHz and 20 MHz
  - 4/4 = 1 MHz instruction cycle = 1/1 MHz = 1 μs
  - 20/4 = 5 MHz instruction cycle = 1/5 MHz = 0.2 μs

Pipelining





D = Decode the instruction

R = Read the operand

P = Process (eg. ADDLW)

W = Write the result to destination register

Reference

- M.A. Mazidi, R.D. Mckinlay, D Causey, PIC Microcontroller and Embedded Systems Using Assembly and C for PIC18, Pearson Education Inc., 2008.
- Han-Way Huang, PIC Microcontroller: An Introduction to Software and Hardware Interfacing, Thomson Delmar Learning, 2005.