CP316
Assembly Language Programming

Terry Sturtevant

Wilfrid Laurier University

November 8, 2017

Introduction to the PIC Assembly Language

Introduction to the PIC Assembly Language

instruction fields

Introduction to the PIC Assembly Language

instruction fields

→ Section 2.3

Introduction to the PIC Assembly Language

instruction fields

→ Section 2.3

→ Section 20.0

language operands

Introduction to the PIC Assembly Language

instruction fields

→ Section 2.3

→ Section 20.0

language operands

→ Section 1.10

Introduction to the PIC Assembly Language

instruction fields

→ Section 2.3

→ Section 20.0

language operands

→ Section 1.10

→ Section 1.8

Introduction to the PIC Assembly Language

instruction fields

→ Section 2.3

→ Section 20.0

language operands

→ Section 1.10

→ Section 1.8

→ Section 1.9

Introduction to the PIC Assembly Language

instruction fields

→ Section 2.3

→ Section 20.0

language operands

→ Section 1.10

→ Section 1.8

→ Section 1.9

assembler directives

Introduction to the PIC Assembly Language

instruction fields

→ Section 2.3

→ Section 20.0

language operands

→ Section 1.10

→ Section 1.8

→ Section 1.9

assembler directives

→ Section 2.4

Introduction to the PIC Assembly Language

instruction fields

→ Section 2.3

→ Section 20.0

language operands

→ Section 1.10

→ Section 1.8

→ Section 1.9

assembler directives

→ Section 2.4

assembler directives vs. language instructions

assembler directives vs. language instructions

assembler directives vs. language instructions

- **assembler directives**

assembler directives vs. language instructions

- **assembler directives**

  for the *compiler*

assembler directives vs. language instructions

- **assembler directives**

  for the *compiler*

  do not become lines of code

assembler directives vs. language instructions

- **assembler directives**

  for the *compiler*

  do not become lines of code
- **language instructions**

assembler directives vs. language instructions

- **assembler directives**

  for the *compiler*

  do not become lines of code
- **language instructions**

  for the *microprocessor*

assembler directives vs. language instructions

- **assembler directives**

  for the *compiler*

  do not become lines of code
- **language instructions**

  for the *microprocessor*

  become lines of code

sample code

sample code

```
here  andwf  SSPCON2,W ;stuff
      addwf  TMR0LCOPY,F,A
```

Instruction fields

Instruction fields

- **here**

Instruction fields

- **here**

  optional label

Instruction fields

- **here**

  optional label
- **andwf**

Instruction fields

- **here**

  optional label
- **andwf**

  command

Instruction fields

- **here**

  optional label
- **andwf**

  command
- **SSPCON2**

Instruction fields

- **here**

  optional label
- **andwf**

  command
- **SSPCON2**

  file register

Instruction fields

- **here**

  optional label
- **andwf**

  command
- **SSPCON2**

  file register
- **,F**

Instruction fields

- **here**

  optional label
- **andwf**

  command
- **SSPCON2**

  file register
- **,F**

  destination (F or W)

Instruction fields

- **here**

  optional label
- **andwf**

  command
- **SSPCON2**

  file register
- **,F**

  destination (F or W)
- **,A**

Instruction fields

- **here**

  optional label
- **andwf**

  command
- **SSPCON2**

  file register
- **,F**

  destination (F or W)
- **,A**

  access bank; i.e. no BSR

Instruction fields

- **here**

  optional label
- **andwf**

  command
- **SSPCON2**

  file register
- **,F**

  destination (F or W)
- **,A**

  access bank; i.e. no BSR
- **;**

Instruction fields

- **here**

  optional label
- **andwf**

  command
- **SSPCON2**

  file register
- **,F**

  destination (F or W)
- **,A**

  access bank; i.e. no BSR
- **;**

  comment after this

Program Template

Program Template

sample organization

Program Template

sample organization

→ Section 2.6

Program Template

sample organization

→ Section 2.6

→ Section 1.5.4

Program Template

sample organization

→ Section 2.6

→ Section 1.5.4

looping

Program Template

sample organization

→ Section 2.6

→ Section 1.5.4

looping

→ Section 2.9

Program Template

sample organization

→ Section 2.6

→ Section 1.5.4

looping

→ Section 2.9

meaning of “end” directive

Program Template

sample organization

→ Section 2.6

→ Section 1.5.4

looping

→ Section 2.9

meaning of “end” directive

→ Section 2.11 (bad example)

Program Template

sample organization

→ Section 2.6

→ Section 1.5.4

looping

→ Section 2.9

meaning of “end” directive

→ Section 2.11 (bad example)

Instructions

Instructions

addressing modes

Instructions

addressing modes

→ Section 1.9

Instructions

addressing modes

→ Section 1.9

FSRs

Instructions

addressing modes

→ Section 1.9

FSRs

sample instructions

Instructions

addressing modes

→ Section 1.9

FSRs

sample instructions

→ Section 1.10

Instructions

addressing modes

→ Section 1.9

FSRs

sample instructions

→ Section 1.10

result destination operands

Branch Instructions

Branch Instructions

bra vs. **goto**

Branch Instructions

bra vs. **goto**

→ Section 2.9.2

Branch Instructions

bra vs. **goto**

→ Section 2.9.2

$ = current instruction

Branch Instructions

bra vs. **goto**

→ Section 2.9.2

$ = current instruction

1 word instructions, except for **movff**, **goto**, **call**,**lfsr**

Branch Instructions

bra vs. **goto**

→ Section 2.9.2

$ = current instruction

1 word instructions, except for **movff**, **goto**, **call**,**lfsr**

skip instructions

Subroutines

Subroutines

call vs. **rcall**

Subroutines

call vs. **rcall**

→ Section 2.9.2

Subroutines

call vs. **rcall**

→ Section 2.9.2

return vs. **retlw**

Subroutines

call vs. **rcall**

→ Section 2.9.2

return vs. **retlw**

hardware stack

Subroutines

call vs. **rcall**

→ Section 2.9.2

return vs. **retlw**

hardware stack

→ Section 1.5.4

Subroutines

call vs. **rcall**

→ Section 2.9.2

return vs. **retlw**

hardware stack

→ Section 1.5.4

→ Section 4.7

Subroutines

call vs. **rcall**

→ Section 2.9.2

return vs. **retlw**

hardware stack

→ Section 1.5.4

→ Section 4.7

subroutines vs. macros

Subroutines (continued)

Subroutines (continued)

shadow registers

Subroutines (continued)

shadow registers

→ Section 4.7.1

Subroutines (continued)

shadow registers

→ Section 4.7.1

fast register stack

Subroutines (continued)

shadow registers

→ Section 4.7.1

fast register stack

→ Section 4.7.5

Subroutines (continued)

shadow registers

→ Section 4.7.1

fast register stack

→ Section 4.7.5

return FAST

Subroutines (continued)

shadow registers

→ Section 4.7.1

fast register stack

→ Section 4.7.5

return FAST

Data in Program Memory

Data in Program Memory

tblrd vs. **lfsr**

Data in Program Memory

tblrd vs. **lfsr**

→ Section 2.10

Data in Program Memory

tblrd vs. **lfsr**

→ Section 2.10

tblptr**l**, tblptr**h**, tblptr**u**

Data in Program Memory

tblrd vs. **lfsr**

→ Section 2.10

tblptr**l**, tblptr**h**, tblptr**u**

Other Instructions

Other Instructions

logical instructions

Other Instructions

logical instructions

→ Section 2.11

Other Instructions

logical instructions

→ Section 2.11

rotate instructions; (**nc** vs. **c**)

Other Instructions

logical instructions

→ Section 2.11

rotate instructions; (**nc** vs. **c**)

→ Section 2.13

Timing Loops

Timing Loops

instruction time

Timing Loops

instruction time

→ Section 1.7