

# Chapter 09

## Programming in Assembly

# Lesson 02

**Assembly language program and  
Program steps for the basic  
instructions**

# ALP program Lines

1. Header Lines
2. List of Called Routines
3. Required initial hardware, conditions and calling conditions
4. Constants and RAM variables assignments using EQU

# ALP program Lines

5. Main program origin address
6. Main program codes in mnemonics
7. Routine origin address
8. Routine program codes in mnemonics

# ALP program Lines

9. ISR program origin address

10. ISR program codes in mnemonics

11. Table origin address and table data assignments

12. String origin address and string characters

# ALP program Lines

13. Interrupt vector origin

14. Reset vector origin

15. Watchdog timer reset origin

16. String origin address and string characters

# Use of Comments

- Essential part of a program
  - Semicolon - symbol for comments start

# Example of comments describing required initial hardware, conditions and calling conditions

; 8051 Learning kit- 8751 12MHz

; External RAM 0000-3FFFFH

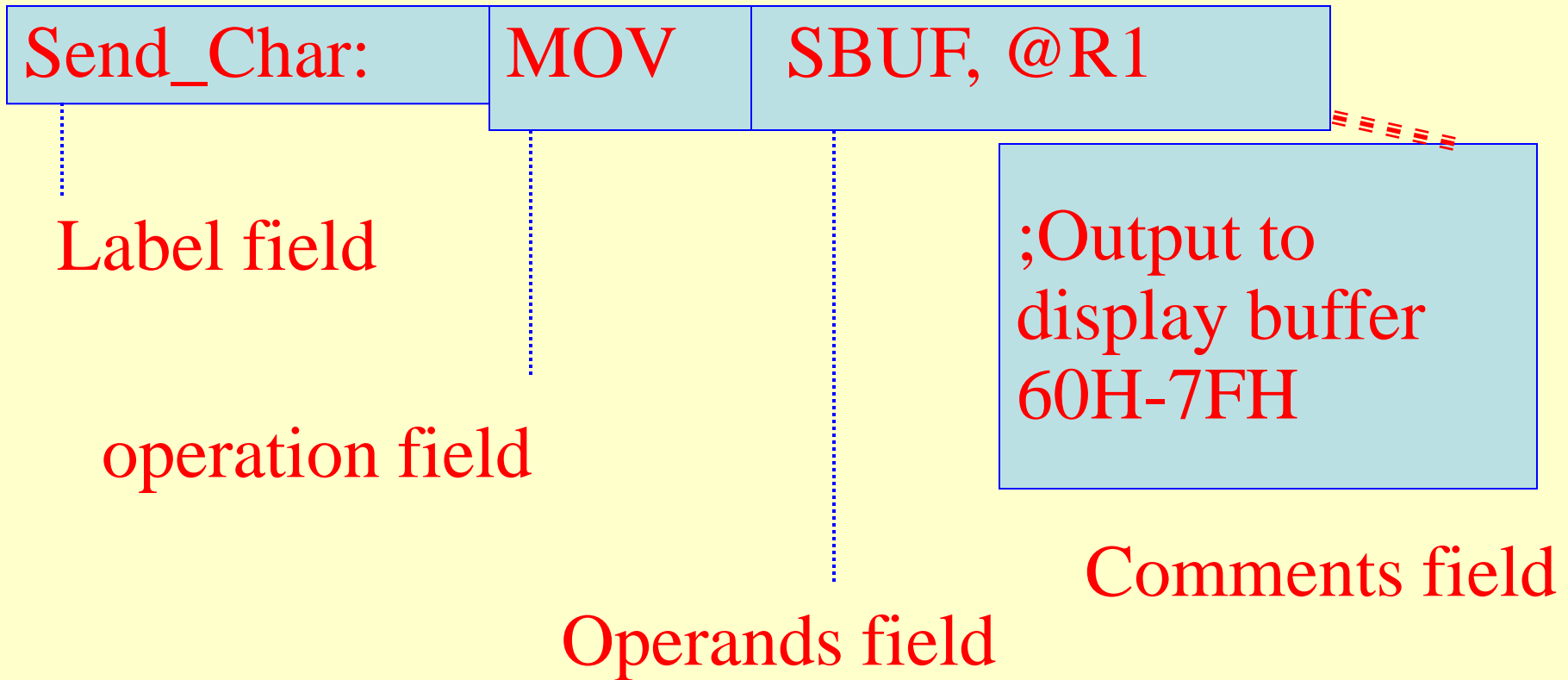
; PROM Internal 4kB, SI enabled



## Header Lines 1-4 in ALP

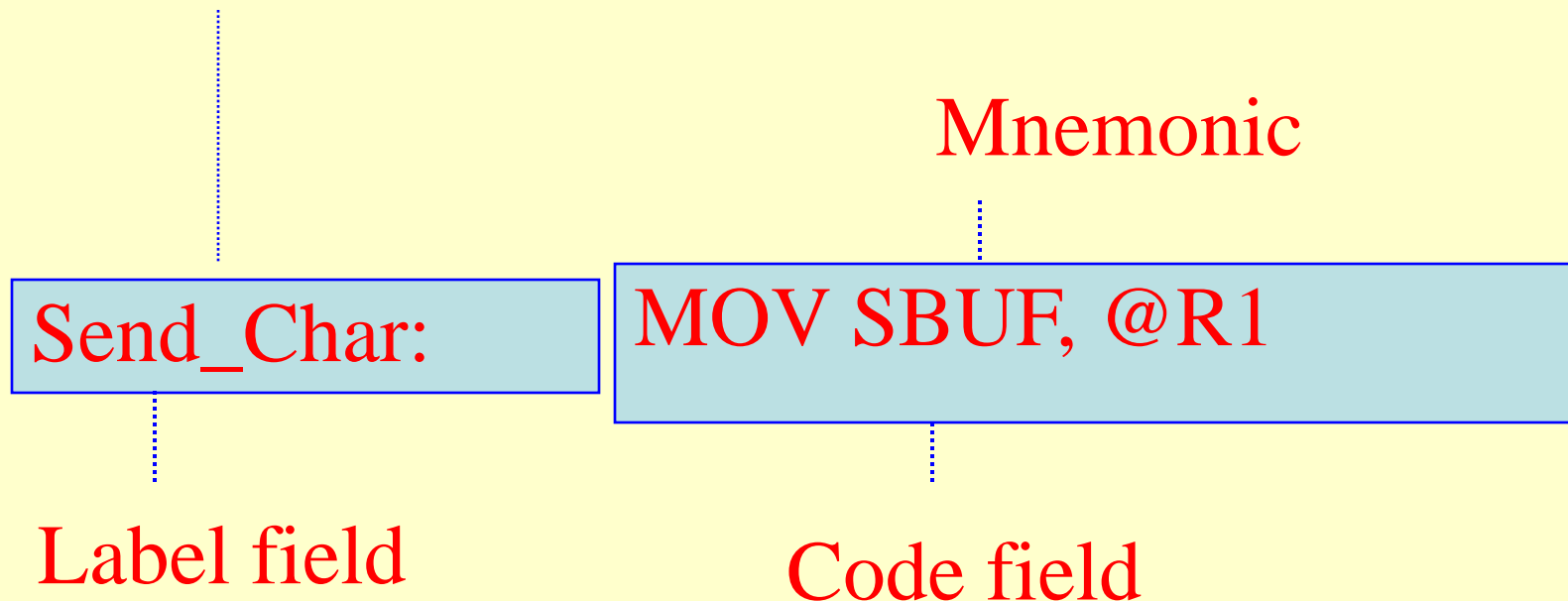
1. Program name, version,  
; Send\_Characters,version 1.0
2. Programmer name, date  
; Raj-Kamal July 17, 2011
3. Brief description  
; Input from SBUF (serial UART)  
;Output to display buffer 60H-7FH

# Four Fields in an ALP line



# Line for Main Program Origin

A suitable start name as label or address:



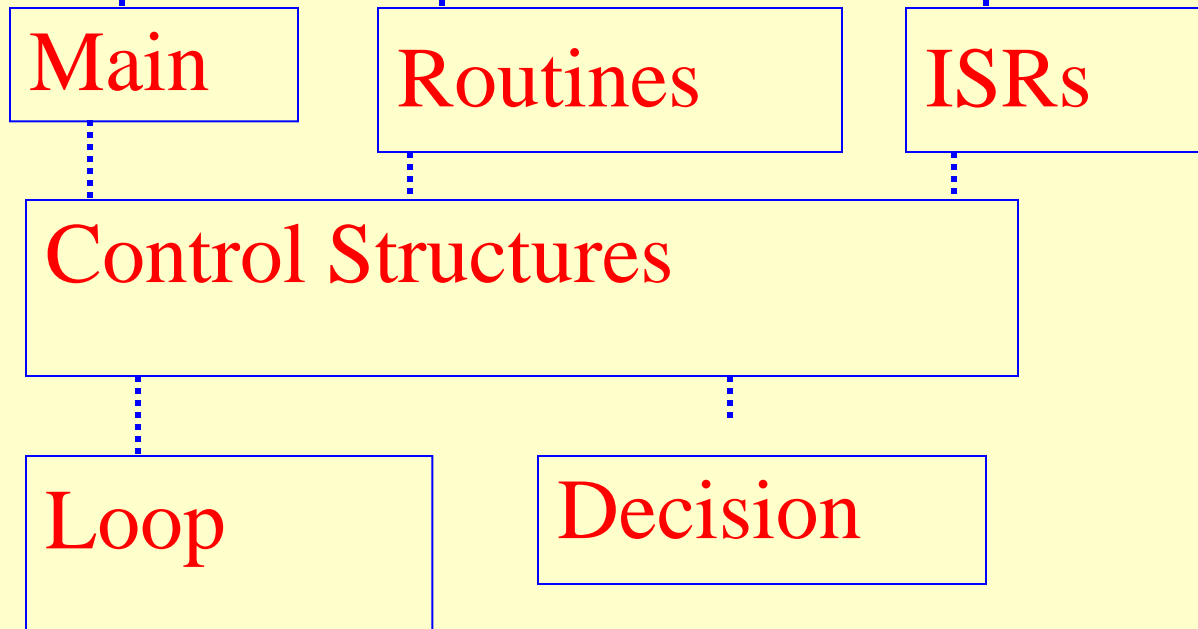
# Fields at an Assembler Listing Line

**LineNum**      **Address**      **Codes**

<b>Label</b>	<b>Opcode</b>	<b>Operands</b>	<b>Comments</b>
--------------	---------------	-----------------	-----------------

# Assembler Addresses Allocation

Program Code-blocks



# Assembler Data Sets

Variables,  
Tables,  
Stacks,  
Message  
Strings,  
Queues

# Data (Values) Assignment by Assembler

variables

Expressions

Constants

Tables, Stacks, Message Strings, Queues

# Assembler Directives

- Fifty directives in new generation assemblers
- Refer Table 9.1 gives the important and common directives



# Directives in an Assembler

- Module control
- Symbol control
- Value assignment
- Conditional control and data and code segments control

# Exemplary Assembler Directives

**ORG**

Assign present location as origin

**EQU**

Define a variable value

**SET**

Set a variable

**DB**

Save data bytes

**DS**

Allocate data storage

# Assembler Types

# Absolute Assembler

Generates Codes, which can be directly put at the device (MCU) addresses in an .abs file

# Reallocatable Assembler

Generates Codes, which can be reallocated before put at the device (MCU) addresses for an .abs file

Reallocates when the segment address changes or assembled code-blocks or files added

# Structured Assembler

1. Use of control structures
2. Use of segment control
3. Use of structures enables simpler and faster coding development

# Forward reference Two pass Assembler

Permits use of symbols without address allocation by the programmer

First pass traces and records the symbols found

Second pass allocation of addresses

# Macro Assembler

1. Use of Macros
2. Use of a Macro processing language



# Cross-Assembler

1. Generates code for host from the assembled codes
2. Generates code for MCU from the codes tested at the host

# Summary

# We learnt

- Assembly Language Program
- Fields- label, operation, operands and comments
- When the ALP required
- Assembler helps in many ways
- Use of macros, directives, expressions, and control structures

# We learnt

- Type of Assemblers

End of Lesson 02 on

**Assembly language program and  
Program steps for the basic  
instructions**