

SOFTMAX

ONLINE SCHOOL

Haque Villa, Rangamati Nir, DUET, Gazipur-1707

01784450949, 0967 8677 677

sosbd24@gmail.com

অধ্যায়-৬

PIC Mid Range মাইক্রোকন্ট্রোলারের অ্যাসেম্বলি ল্যাংগুয়েজ প্রোগ্রামিং

❖ Assembly Language হচ্ছে Machine Language এর পরবর্তী প্রোগ্রামের ভাষা। এ Language এর প্রচলন শুরু হয় ১৯৫০ সাল থেকে। দ্বিতীয় প্রজন্মের কম্পিউটারে এ ভাষা ব্যাপকভাবে প্রচলিত ছিল। Assembly Language এ বিভিন্ন সংকেত বা নিমোনিক (Mnemonic) ব্যবহার করে প্রোগ্রাম লেখা হয়।

যেমন : যোগের জন্য সংকেত = ADD

বিয়োগের জন্য সংকেত = SUB

গুণের জন্য সংকেত = MUL

ভাগের জন্য সংকেত = DIV ইত্যাদি।

Assembly Language এর সুবিধাসমূহ :

- Assembly Language সহজে বুঝা যায়।
- এ ভাষায় প্রোগ্রাম রচনা করা Machine Language এর তুলনায় সহজ।
- প্রোগ্রাম এর আকার তুলনামূলক ছোট।
- Memory Address বিবরণের প্রয়োজন হয় না।

Assembly Language এর অসুবিধা :

- এক মেশিনের জন্য লেখা প্রোগ্রাম অন্য মেশিনে ব্যবহার করা যায় না।
- Program Execution এ Translator Program হিসেবে Assembler এর প্রয়োজন হয়।
- নিমোনিকসমূহ মুখস্ত রাখতে হয়।
- প্রোগ্রাম রচনা করা অত্যন্ত ক্লান্তিকর ও সময় সাপেক্ষ।

❖ **Assembler:** অ্যাসেম্বলারের সাহায্যে অ্যাসেম্বলি ল্যাংগুয়েজে লেখা প্রোগ্রামকে মেশিন কোডে রূপান্তর করা হয়। অবজেক্ট কোডকে বা বাইনারি কোডকে OBJ এক্সটেনশন যুক্ত ফাইলে সংরক্ষণ করা হয়। এই ফাইলকে অবজেক্ট ফাইল বলা হয়। অবজেক্ট ফাইল ছাড়াও অ্যাসেম্বলারের সাহায্যে প্রোগ্রামের লিস্ট ফাইল তৈরি করা হয়। এই ফাইলে LST এক্সটেনশন ব্যবহৃত হয়। লিস্ট ফাইলে প্রোগ্রামের অ্যাসেম্বলি কোড এবং মেশিন কোড থাকে।

Assembly Language প্রোগ্রামকে মেশিন কোডে রূপান্তরের জন্য বিভিন্ন ধরনের Assembler ব্যবহার করা হয়।

যেমন :

- One-Pass Assembler
- Two-Pass Assembler
- Macro Assembler
- Meta Assembler

*** ১। অ্যাসেম্বলি ভাষা (Assembly Language) কী?

বাকশিবো- ২০১৭

উত্তর : বিশেষ ধরনের নিমোনিক (ADD, SUB, MOV, MUL, DIV) ইত্যাদি ব্যবহার করে যে Program তৈরি করা হয়, তাকে Assembly Language বলে।

যেমন, MASM (Microsoft Micro Assembler), TASM (Turbo Assembler) ইত্যাদি।

*** ২। অ্যাসেম্বলার (Assembler) কাকে বলে?

বাকশিবো- ২০১৬'পরি

উত্তর : যে অনুবাদক Program, Assembly Language এ লিখিত Program কে Machine Language এ রূপান্তর করে, তাকে Assembler বলে।

** ৩। অবজেক্ট ফাইল (Object File) কাকে বলে?

বাকশিবো- ২০১৮'পরি

উত্তর : Assembler এর সাহায্যে Assembly Language এ লেখা প্রোগ্রামকে Machine Code বা Object Code এ রূপান্তর করা হয়, যার Extension obj যুক্ত। এই ফাইলকে Object File বলা হয়। এই File 0 এবং 1 নিয়ে গঠিত এবং মেশিন এই সকল ফাইল সরাসরি বুঝতে পারে।

** ৪। লিংকার (Linker) কী?

উত্তর : একটি বড় প্রোগ্রামকে ছোট ছোট প্রোগ্রাম মডিউলে লেখা হয়। ছোট ছোট মডিউলগুলোকে আলাদা আলাদাভাবে লেখা হয়, পরীক্ষা করা হয় এবং ডিবাগিং করা হয়। ফলে প্রত্যেকটি মডিউলের জন্য আলাদা আলাদা অবজেক্ট ফাইল উৎপন্ন হয়। লিংকার এর সাহায্যে একাধিক অবজেক্ট ফাইলকে একটি বড় অবজেক্ট ফাইলে সংযুক্ত করা হয়।

*** ৫। IDE বলতে কী বুঝায়?

বাকশিবো- ২০১৪, ২০

উত্তর : IDE এর পূর্ণরূপ Integrated Development Environment. ইহা এমন এক ধরনের Environment, যা কম্পিউটার প্রোগ্রামারদের Software তৈরিতে সহায়তা করে। এখানে একজন প্রোগ্রামার Source Code লিখতে পারে, Compile করতে পারে এবং Debugger দ্বারা ভুল সংশোধন করতে পারে।

* ৬। Instruction বা নির্দেশনা বলতে কী বুঝ?

উত্তর : Instruction হচ্ছে একটি কমান্ড, যার সাহায্যে মাইক্রোকন্ট্রোলারকে কোনো কাজ করার নির্দেশ প্রদান করা হয়।

*** ৭। Instruction Set বলতে কী বুঝ?

বাকশিবো- ২০১৪, ১৭, ২০

উত্তর : Instruction Set হলো মেশিন কোড ইনস্ট্রাকশনের সমষ্টি, যা প্রসেসরকে বিভিন্ন ধরনের কার্য নির্বাহ করায়। কোনো প্রসেসর কর্তৃক বোধগম্য সকল Instruction সমূহকে একত্রে Instruction Set বলা হয়।

যেমন : Base Line Microcontroller এর Instruction Set 33 টি

Mid Range Microcontroller এর Instruction Set 35 টি

PIC 18 Microcontroller এর Instruction Set 89 টি ইত্যাদি।

* ৮। Mid Range PIC Microcontroller এর Instruction কে কয়ভাগে ভাগ করা হয়?

উত্তর : Mid Range PIC Microcontroller এর Instruction কে তিন ভাগে ভাগ করা হয়। যথা :

- Byte Oriented File Register Operation
- Bit Oriented File Register Operation এবং
- Literal and Control Operation

** ৯। Assembly Language এর Field গুলোর নাম লেখ।

বাকশিবো- ২০১৯

উত্তর : Assembly Language এর Field চারটি। যথা :

- Label (লেবেল)
- Opcode (অপকোড)
- Operand (অপারেণ্ড)
- Comment (কমেন্ট)

** ১০। CPU Instruction কাকে বলে?

বাকশিবো- ২০১৪

উত্তর : Instruction হচ্ছে একটি কমান্ড, যার সাহায্যে মাইক্রোপ্রসেসরকে কোনো কাজ করার নির্দেশ প্রদান করা হয়। Assembly Language এ প্রতিটি Statement, একটি CPU Instruction এর সাথে সামঞ্জস্যপূর্ণ।

*** ১১। অ্যাসেম্বলার ডিরেকটিভ (Assembler Directive) কাকে বলে?

বাকশির্বো- ২০১৬, ১৭'পরি, ১৮, ১৯'পরি

উত্তর : অ্যাসেম্বলার ইনস্ট্রাকশন বা অ্যাসেম্বলার সুডো ইনস্ট্রাকশন হচ্ছে অ্যাসেম্বলারের প্রতি বিশেষ ধরনের নির্দেশ, যা সোর্স প্রোগ্রামে লিখিত এক ধরনের স্টেটমেন্ট বা ইনস্ট্রাকশন, যা মেশিন কোডে রূপান্তর করা হয় না।

* ১২। এডিটর প্রোগ্রাম (Editor Program) কী?

উত্তর : এডিটরের সাহায্যে অ্যাসেম্বলি প্রোগ্রামের সোর্স ফাইল তৈরী করা হয়। এই সোর্সে ফাইল হতে অ্যাসেম্বলারের সাহায্যে অবজেক্ট ফাইল তৈরী করা হয়। প্রোগ্রামে কোনো ভুল থাকলে এডিটরের সাহায্যে সোর্স ফাইলকে এডিট করা যায়। এছাড়া এডিটরের সাহায্যে সোর্স ফাইলকে সেকেন্ডারী মেমোরীতে সংরক্ষণ করা হয়, যাতে সোর্স ফাইলকে পরবর্তিতে ব্যবহার করা যায়।

** ১৩। Software Development Tools কী?

বাকশির্বো- ২০১৮

উত্তর : অ্যাসেম্বলি ল্যাংগুয়েজে লিখিত প্রোগ্রামকে ডেভেলপ করার জন্য বিভিন্ন ধরনের সাহায্যকারী প্রোগ্রাম ব্যবহৃত হয়। এ সকল সাহায্যকারী প্রোগ্রামকে প্রোগ্রাম ডেভেলপমেন্ট টুলস বলা হয়। যেমন : অ্যাসেম্বলার, এডিটর, লিংকার, লোকেটর, ডিবাগার, ইমুলেটর ইত্যাদি।

* ১৪। W Register কী?

উত্তর : W Register কে Working Register বলা হয়। যেকোনো Data এই Register এ সরাসরি Read-Write করা যায়। বেশিরভাগ Microcontroller এ Working Register হিসেবে Accumulator ব্যবহৃত হয়।

SOS সংক্ষিপ্ত প্রশ্নোত্তর :

* ১। Assembly Language এর সুবিধা ও অসুবিধাগুলো লেখ।

উত্তর : Assembly Language এর সুবিধাসমূহ :

- Assembly Language সহজে বুঝা যায়।
- এ ভাষায় প্রোগ্রাম রচনা করা Machine Language এর তুলনায় সহজ।
- প্রোগ্রাম এর আকার তুলনামূলক ছোট।
- Memory Address বিবরণের প্রয়োজন হয় না।

Assembly Language এর অসুবিধা :

- এক মেশিনের জন্য লেখা প্রোগ্রাম অন্য মেশিনে ব্যবহার করা যায় না।
- Program Execution এ Translator Program হিসেবে Assembler এর প্রয়োজন হয়।
- নিমোনিকসমূহ মুখস্ত রাখতে হয়।
- প্রোগ্রাম রচনা করা অত্যন্ত ক্লান্তিকর ও সময় সাপেক্ষ।

২। Assembly Language এর ব্যবহার বা প্রয়োগক্ষেত্র লেখ।

উত্তর : Assembly Language এর ব্যবহার নিম্নরূপ :

- Driver Software তৈরিতে
- Operating System এ
- Game Development এ
- Embedded System এ
- Security Research ইত্যাদি।

*** ৩। অ্যাসেম্বলি ভাষার Field সমূহ বর্ণনা কর।

উত্তর : অ্যাসেম্বলি ল্যাংগুয়েজ প্রোগ্রাম লেখার ক্ষেত্রে সাধারণত চারটি ফিল্ড ব্যবহার করে প্রোগ্রাম লেখা হয়। ব্যবহৃত চারটি ফিল্ড হলো :

- লেবেল ফিল্ড (Label field)
- অপ-কোড ফিল্ড (OP-code field)
- অপারেণ্ড ফিল্ড (Operand field) এবং
- কমেন্ট ফিল্ড (Comment field).

১) লেবেল ফিল্ড : এ ফিল্ডের মাধ্যমে প্রোগ্রামের মধ্যে লেবেল ব্যবহার করা যায়। যেমন : সাবরুটিনের শুরুতে লেবেল ব্যবহার করতে হয়। কারণ সাবরুটিনটিকে এর প্রথম ইনস্ট্রাকশনের লেবেলের নাম কল (CALL) করা হয়। মূলত এই লেবেল হলো একটি অ্যাড্রেস। লেবেলের সাথের ইনস্ট্রাকশনটির অপ-কোড মেমরির যে লোকেশনে থাকে, সেই অ্যাড্রেসটি হলো উক্ত লেবেল।

২) অপ-কোড ফিল্ড : অপ-কোড ফিল্ডে অ্যাসেম্বলি ল্যাংগুয়েজের অপারেশন কোড (Operation code) বা অপ-কোড (OP-code) লেখা হয়। অপ-কোডের সাহায্যে মাইক্রোপ্রসেসরকে কাজের নির্দেশ দেয়া হয়। অপ-কোড ফিল্ড অ্যাসেম্বলি ল্যাংগুয়েজের একটি অপরিহার্য ফিল্ড। প্রদত্ত উদাহরণে “MOV” অপ-কোডটি অপ-কোড ফিল্ডে অবস্থিত।

৩) অপারেণ্ড ফিল্ড : অপারেণ্ড ফিল্ডে অ্যাসেম্বলি ল্যাংগুয়েজের অপারেণ্ডসমূহ লেখা হয়। অপারেণ্ড হলো একটি ইনস্ট্রাকশনের অপ-কোডের পরের অংশ। অপারেণ্ড হিসেবে সাধারণত রেজিস্টারের নাম, মেমরি অ্যাড্রেস, লেবেলের নাম, ইমিডিয়েট (Immediate/literal) ডাটা ইত্যাদি ব্যবহৃত হয়। অপ-কোডের মাধ্যমে মাইক্রোপ্রসেসরকে কোনো অপারেশন করার নির্দেশ করা হয়।

৪) কমেন্ট ফিল্ড : এই ফিল্ডটি একেবারেই ঐচ্ছিক ফিল্ড। প্রোগ্রামার নিজের সুবিধার্থে কমেন্ট ফিল্ডের মধ্যে কমেন্ট (মন্তব্য, টিকা) ব্যবহার করতে পারে। কমেন্ট ফিল্ডের মধ্যে সাধারণত ইনস্ট্রাকশন অনুযায়ী ইনস্ট্রাকশনের কাজকে সংক্ষেপে লেখা হয়। বড় বড় প্রোগ্রামে এর ব্যবহার সুবিধাজনক। কমেন্ট ব্যবহার করতে হলে কমেন্টের শুরুতেই সেমিকোলন চিহ্ন (;) দিতে হয়।

**** ৪। অ্যাসেম্বলি ভাষার অপারেণ্ড হিসেবে কী কী ব্যবহৃত হয়?**

বাকশির্বো- ২০২০

উত্তর : অ্যাসেম্বলি ভাষার অপারেণ্ড হিসেবে নিম্নোক্ত জিনিসসমূহ ব্যবহৃত হয় :

- Immediate বা Literal Data
- CPU এর কোনো Register
- Memory Location বা Address
- কোনো Label এর নাম
- Input Output Part

**** ৫। PIC Microcontroller এর Assembler Directive সমূহ বর্ণনা কর।**

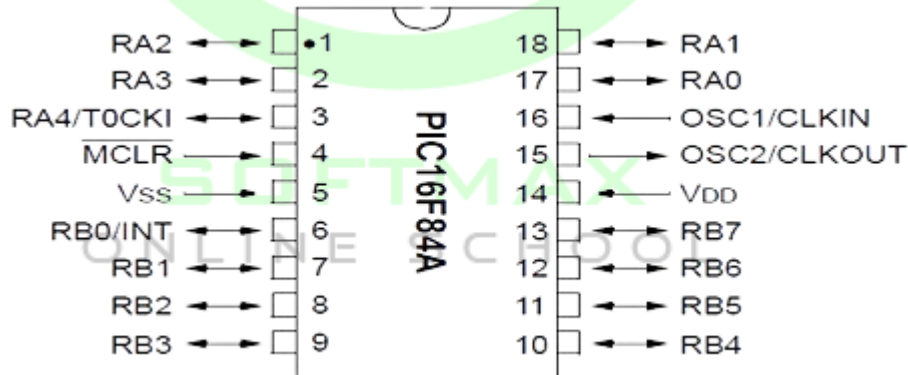
উত্তর : অ্যাসেম্বলার ডাইরেক্টিভ বা অ্যাসেম্বলার সুডো ইনস্ট্রাকশন হচ্ছে অ্যাসেম্বলারের প্রতি বিশেষ ধরনের নির্দেশ, যা সোর্স প্রোগ্রামে লিখিত এক ধরনের স্টেটমেন্ট বা ইনস্ট্রাকশন, যা মেশিন কোডে রূপান্তর করা হয় না।

PIC Microcontroller এর Assembler Directive সমূহ নিম্নরূপ :

- ORG: ORG এর পূর্ণরূপ Origin. Program এর Code কোন Location হতে Load হবে তা নির্ধারণ করাই মূলত ORG এর কাজ।
যেমন : ORG 100H
- EQU: EQU এর পূর্ণরূপ Equate. কোনো Constant মান বা Fixed Address কে নির্ধারণ করতে এই Directive ব্যবহৃত হয়।
যেমন : NUM1 EQU 12H
- INCLUDE: Header file সংযুক্ত করতে এই Directive ব্যবহৃত হয়।
যেমন : #INCLUDE PIC 16F877A.INC
- END: অ্যাসেম্বলি প্রোগ্রামের সমাপ্তি বুঝাতে এই Directive ব্যবহৃত হয়। এরপর যদি আরও কোনো লাইন থাকেও, তবে সে সকল লাইন আর Execute হয় না। এছাড়াও নির্দিষ্ট PIC Microcontroller বুঝাতে LIST, Hexadecimal হতে Decimal এ রূপান্তর করতে RADIX ব্যবহৃত হয়।

**** ৬। MID Range PIC 16F84A মাইক্রোকন্ট্রোলারের পিন ডায়াগ্রাম অঙ্কন কর।**

উত্তর :



Mid Range PIC 16F84A মাইক্রোকন্ট্রোলার 18 পিনের PDIP Package (PDIP = Plastic Dual In line Package). যার দুইটি Port থাকে Port (RA0 -RA4) এবং port B (RB0 -RB7)

V_{DD} = Positive Power Supply

V_{SS} = Ground

OSC1 = Clock Input

OSC2 = Clock Output

MCLR = Master Clear, যা Reset করার জন্য ব্যবহৃত হয়।

*** ১। PIC Microcontroller এর ইনস্ট্রাকশনের শ্রেণিবিভাগ বর্ণনা কর।

উত্তর : Mid Range PIC Microcontroller এর Instruction কে তিন ভাগে ভাগ করা হয়। যথা :

- Byte Oriented File Register Operation
- Bit Oriented File Register Operation এবং
- Literal and Control Operation

i) **Byte Oriented file Register Operation:**

Format:

13	8	7	6	0
OPCODE	d(BIT#)	f(FILE#)		

14 bit Instruction কে তিন ভাগে ভাগ করা হয়।

0-6 এই 7bit = File Register Address

d = Destination (1 bit)

যদি d = 0 হয়, তবে Destination W (Working Register)

d = 1 হয়, তবে Destination f (file Register)

8-13 = 6 bit = OPcode (Operation Code)

f এর মান 0-127 ($2^7 = 128$)

d এর মান 0, অথবা 1(1 bit)

Mnemonic, Operands	বর্ণনা	সাইকেল
ADDWF f,d	W এর সাথে f এর মান যোগ করে।	1
ANDWF f,d	W এর সাথে f এর মানের প্রতিটি বিটের Logical AND Operation করে।	1
CLRF f	f এর মানকে Clear (Reset) করে।	1
CLRWF	W এর মানকে Clear (Reset) করে।	1
COMF f,d	f এর মানকে Decrement বা হ্রাস করে।	1
DECF f,d	f এর মানকে Decrement বা হ্রাস করে।	1
DECFSZ f,d	f এর মানকে Decrement বা হ্রাস করে, যদি মান 0 হয় তবে পরের লাইন গুলো নির্বাহ হয় না (Skip) করে।	1 (2)
INCF f,d	f এর মানকে Increment বা বৃদ্ধি করে।	1
INCFSZ f,d	f এর মানকে Increment বা বৃদ্ধি করে, যদি মান 0 হয়, তবে পরের লাইন গুলো নির্বাহ হয় না (Skip) করে।	1 (2)
IORWF f,d	W এর মানের সাথে f এর মানের প্রতিটি বিট Logical OR Operation করে।	1
MOVF f,d	f এর মানকে স্থানান্তর বা Copy করে।	1
MOVWF f	W এর মানকে f এ স্থানান্তর বা Copy করে।	1
NOP	No Operation	1
RLF f,d	f কে Carry সহ Rotate left করে।	1
RRF f,d	f কে Carry সহ Rotate Right করে।	1
SUBWF f,d	W থেকে f এর মানকে বিয়োগ করে।	1
SWAPF f,d	f এর অভ্যন্তরীণ চার bit Swap করে।	1
XORWF f,d	W এর সাথে f এর মানের প্রতিটি বিটের Logical XOR Operation হয়।	1

ii) **Bit Oriented file Register Operations:**

From:	13	10 9	7 6	0
	OP Code	b(BIT#)	f(FILE#)	

f = 7 bit, এর মান $2^7 = 128$ (0 – 127) File Address

b = 3 bit, এর মান $2^3 = 8$ (0 – 7) bit Address

Opcode = 4 bit

Mnemonic, Operation	বর্ণনা	সাইকেল
BCF f,d	f এর নির্দিষ্ট একটি bit Clear (0) করে।	1
BSF f,d	f এর নির্দিষ্ট একটি bit Set (1) করে।	1
BTFSC f,d	f এর নির্দিষ্ট বিটকে Test করে, যদি Clear হয়, তবে Skip করবে।	1 (2)
BTFSS f,d	f এর নির্দিষ্ট বিটকে Test করে, যদি Set হয়, তবে Skip করবে।	1 (2)

iii) **Literal and Control Operations:**

Format:	13	8 7	0
	OP Code	k(Literal)	

k = 8bit Immediate Value

Mnemonic, Operation	বর্ণনা	সাইকেল
ADDLW k	Immediate Value কে W এর সাথে যোগ করে।	1
ANDLW k	Immediate Value এর সাথে W এর মানের প্রতিটি বিটের Logical AND Operation হবে।	1
CALL k	Subroutine কল	2
CLRWD	Watch Dog Timer কে Clear করে।	1
GOTO k	নির্দিষ্ট একটি Location এ যায়।	2
IORLW k	Immediate Value এর সাথে W এর মানের প্রতিটি bit এর Logical OR Operation হবে।	1
MOVLW k	Immediate Value কে W এ Copy করে।	1
RETFIE	Interrupt হবে Return করে।	2
RETLW k	Immediate Value সহ W এ Return করে।	2
RETURN	Subroutine হতে Return করে।	2
SLEEP	Processor কে Standby Mode এ রাখা।	1
SUBLW k	Immediate Value হতে W এর মান বিয়োগ করে।	1
XORLW k	Immediate Value এর সাথে W এর মানের প্রতিটি বিটের Logical XOR Operation হবে।	1

SOFTMAX
ONLINE SCHOOL

*** ২। অ্যাসেম্বলি ভাষায় প্রোগ্রাম তৈরি এবং সম্পাদনের ধাপগুলো লেখ।

উত্তরঃ **Editor Program:** Assembly Language এ প্রোগ্রাম লেখার জন্য একটি Editor ব্যবহার করতে হবে। Editor এ লেখা ফাইলকে Source Code বা Source Program বা Source file বলা হয়। এই সকল File এর Extension asm অথবা src হয়ে থাকে। Assembly ভাষায় Source Editor হিসেবে অনেক ধরনের Editor আছে। যেমনঃ MASM, TASM, Notepad, Sublime Text ইত্যাদি।

Assembler Program: Assembly Language এ লেখা Source file কে Machine Code এ রূপান্তর করার জন্য অনুবাদক প্রোগ্রাম হিসেবে Assembler ব্যবহৃত হয়। এই ফাইলের Extension obj হয়ে থাকে। ইহাকে Object file বলা হয়। এই ফাইলে মূলত প্রয়োজনীয় Machine Code, Linker এর তথ্য সমস্ত কিছু থাকে। এছাড়াও ব্যবহারকারীর প্রয়োজনে List file তৈরি করতে হয়। যার বিভিন্ন ধরনের Address এবং Syntax Error থাকে।

Linker Program: একটি বড় প্রোগ্রামকে ছোট ছোট প্রোগ্রাম মডিউলে লেখা হয়। ছোট ছোট মডিউলগুলোকে আলাদা আলাদাভাবে লেখা হয়, পরীক্ষা করা হয় এবং ডিবাগিং করা হয়। ফলে প্রত্যেকটি মডিউলের জন্য আলাদা অবজেক্ট ফাইল উৎপন্ন হয়। লিংকার সাহায্যে একাধিক অবজেক্ট ফাইলকে একটি বড় অবজেক্ট ফাইলে সংযুক্ত করা হয়। এই অংশে Absolute file তৈরি হয়, যার Extension abs হয়ে থাকে। এই ফাইলে Machine Code, Data, Memory Location এবং Debugging information থাকে।

OH Program: Absolute ফাইলকে পরবর্তীতে মাইক্রোকন্ট্রোলারের অ্যাসেম্বলারের সাথে আসা একটি Object to hex converter প্রোগ্রামে প্রয়োগ করা হয়। এটি Absolute ফাইলকে মাইক্রোকন্ট্রোলারের সম্পাদনযোগ্য hex ফাইলে রূপান্তর করে। এই ফাইলের Extension হলো 'hex'। পরে এটিকে মাইক্রোকন্ট্রোলারের Program ROM এ জমা করা (Burn) হয়।

বাকশিবো- ২০১৪, ১৪'পরি, ১৫'পরি, ১৭, ১৮, ১৮'পরি, ১৯'পরি, ২১

