

SPECIALIZAREA : **Calculatoare**

## FISA DISCIPLINEI

### PROGRAMARE ÎN LIMBAJ DE ASAMBLARE

Cod disciplina: COBS201

Număr credite : 4

#### 1. Obiectivele disciplinei

Sa dobandeasca o perceptie clara cu privire la posibilitatile nelimitate de utilizare a resurselor hard ale un calculator folosind limbajul de asamblare.

Sa optimizeze programele pe care le implementeaza in functie de hardul avut la dispozitie si de cerintele de timp in realizarea calculelor respective.

Sa selecteze mijloacele de procesare cele mai performante in ceea ce privește puterea de procesare raportata la un pret scăzut.

#### 2. Rezultatele invatarii (se exprima in obiective masurabile ce fac subiectul evaluarii)

##### a. Cunoștințele generale

In cadrul cursului se dobandesc cunoștințe necesare proiectării și dezvoltării programelor in limbaj de asamblare care implica, atat cunoștințe specifice limbajului de asamblare cat si cunoștințe referitoare la hard-ul microcalculatorului folosit. De asemenea sunt prezente într-o manieră structurată noțiuni privind optimizarea folosirii resurselor hard disponibile precum si a resursei timp.

##### b. Cunoștințele de specialitate

Componentele unui sistem de calcul  
Proiectarea si implementarea programelor in limbaj de asamblare.  
Procesare de semnal.

##### c. Competențele generale

Structura si functionarea unui microprocesor. Programarea in limbaj de asamblare a microprocesoarelor

##### d. Competențele de specialitate

Implementarea aplicatiilor in timp real folosind procesoare de semnal.

##### e. Abilitățile cognitive specifice

Cunoștințe și abilități pentru utilizarea limbajelor, mediilor și tehnologiilor de programare.

Cunoștințe și abilități de analiză de proces.

Cunoștințe și abilități de proiectare și utilizare a echipamentelor de uz general și dedicate.

Cunoștințe și abilități de a întocmi și gestiona execuția de proiecte în domeniul automaticii și informaticii aplicate precum și în domenii conexe (mentenanță).

Cunoștințe și abilități de lucru în echipă, integrare de sisteme și cooperare interdisciplinară.

#### 3. Concordanța cu obiectivele planului de învățământ/specializării

##### a. Contribuția rezultatelor invatarii disciplinei la formarea competentelor specializării

Limbajul de asamblare constituie o solutie tot mai raspandita pentru aplicatii care necesita efectuarea unui volum mare de calcule intr-un timp scurt, precum si valorificarea la maximum a resurselor hard disponibile, optimizandu-se in acest fel costul sistemului respectiv.

##### b. Cerințele disciplinare prealabile

4. Structura activității didactice

CURS ..... 28 ore  
 Lucrări practice ..... 28 ore

5. Prezentarea conținutului disciplinei

a. Curs

<i>Conținutul activității</i>	<i>Nr. de ore</i>
<b>Capitolul 1 – Prezentarea generală a structurii unui microcalculator</b> Introducere. Arhitectura von Neumann. Componentele de bază ale unui sistem de calcul U.C.P. Memoria program. Memoria de date. Interfețe pentru cuplarea perifericelor. Interfața de proces. Magistralele microsistemului.	2
<b>Capitolul 2 – Unitatea centrală și de preluare a microprocesorului Z80.</b> Circuitele pentru decodificarea instrucțiunilor. Registrul de instrucțiuni. Registrele unității centrale. Registrele tampon de date și adrese. Unitatea aritmetico-logică. Descrierea conexiunilor externe ale microprocesorului Z80. Descrierea registrelor UCP. Descrierea ciclurilor mașinii. Exemple.	2
<b>Capitolul 3 – Moduri de adresare ale microprocesorului Z80.</b> Exemple	4
<b>Capitolul 4 – Programarea în limbajul de asamblare al procesorului Z80.</b> Setul de instrucțiuni: Instrucțiuni de transfer, instrucțiuni aritmetice și logice, instrucțiuni de I/O și întreruperi, instrucțiuni de salt. Mecanisme de apel a procedurilor. Exemple.	5
<b>Capitolul 5 – Studiul cererilor de magistrală, a întreruperilor și starea HALT.</b> Întreruperile nemascabile. Întreruperile mascabile. Modul 0 de întrerupere. Modul 1 de întrerupere. Modul 2 de întrerupere. Exemple.	4
<b>Capitolul 6 – Familia procesoarelor de semnal TMS320.</b> Introducere. Procesoare de semnal TMS320C2xx. Generația procesoarelor TMS320C3x. Procesoare TMS320C4x. Generația TMS320C5x. Generația de procesoare TMS320C54x. Generația TMS320C6x. Procesoare TMS320C8x. Generația TMS320AVxxx.	2
<b>Capitolul 7 – Structura hardware a procesorului de semnal TMS320C32.</b> Structura internă. Unitatea centrală de preluare. Unitatea de înmulțire. Unitatea aritmetico-logică. Unități aritmetice auxiliare pentru adrese. Registrele unității centrale de preluare și registrul de stare. Organizarea memoriei interne. Memoria cache. Harta de memorie a procesorului TMS320C32. Sistemul de întreruperi: registrul de validare a întreruperilor CPU/DMA (IE), registrul indicatorilor de întrerupere CPU (IF).	2
<b>Capitolul 8 – Setul de instrucțiuni al procesorului de semnal TMS320C32.</b> Introducere. Tipuri de date: reprezentarea numerelor în virgulă fixă, reprezentarea numerelor în virgulă mobilă. Moduri de adresare: adresarea la registru, adresarea directă, adresarea indirectă, adresarea imediată, adresarea relativă la registrul PC. Instrucțiuni de transfer. Instrucțiuni de preluare cu 2 operanți. Instrucțiuni de preluare cu 3 operanți. Instrucțiuni paralele de transfer și preluare. Instrucțiuni de control. Exemple.	7

b. Lucrări practice

<i>Conținutul activității</i>	<i>Nr. de ore</i>
Structura unui program în limbaj de asamblare. Prezentarea simulatorului software "Z80 Simulator IDE".	2
Realizarea de programe în limbaj de asamblare, care analizate folosind simulatorul Z80 pun în evidență și permit o mai bună aprofundare a aspectelor teoretice prezentate la curs cu privire la: modurile de adresare ale microprocesorului Z80, instrucțiuni de transfer, instrucțiuni aritmetice și logice, instrucțiuni de I/O și întreruperi, instrucțiuni de salt.	10

Sistemul de dezvoltare DSProto 32 cu procesorul de semnal TMS320C32. Prezentare generala. Componenta software pentru dezvoltarea de aplicatii.	4
Realizarea de programe in limbajul de asamblare specific procesorului TMS320C3x care presupune parcurgerea urmatoarelor etape: conceperea algoritmului, scrierea programului in limbajul de asamblare corespunzator procesorului TMS320C32, translatarea fisierului text ce contine programul sursa in fisiere obiect relocabile utilizand TMS320C3x Assembler, linkeditarea fisierelor obiect intr-un singur fisier obiect executabil prin reevaluarea atat a legaturilor interne, cat si a celor externe utilizandu-se un program linkeditor, incarcarea programului in memoria SRAM a platformei DSProto 32, care se realizeaza cu programul DSProto 32 Debugger, rularea programului incarcat in memorie SRAM i, evident, verificarea rezultatului obtinut.	12

## 6. Invatare

### a. Forme de invatare/predare

Experimentul, descoperirea dirijata, problematizarea, studiul de caz, demonstratia, dialogul, inductia

### b. Resurse educationale

Suport de curs disponibil atat la biblioteca cat si in format electronic

### c. Bibliografie disponibila:

1.M. Patrubany, *Totul despre... microprocesorul Z80*, Editura Tehnică București, 1989.

2.Gh. Toaese, *Introducere in microprocesoare*, Ed. Stiintifica si enciclopedica București, 1986

3.Gh. Pușcașu, V. Nicolau, V. Puscașu, B. Codreș, *Procesoare de semnal TMS320C3x. Aplicatii* Ed. Academica: 2004

### d. Alte resurse

Hardware: calculatoare, DSP-uri, platforme laborator echipate cu microprocesorul Z80.

Software: Z80 Simulator IDE, asamblare DSP.

## 7. Studiu individual: 44 ore

## 8. Evaluare

### a. Forme de evaluare

A) Evaluare cu caracter sumativ

Examen final scris: teorie, intrebari și probleme.

B) Evaluare cu caractere diagnostic și formativ

Examinarea periodica in cadrul orelor de laborator pentru stabilirea modului de asimilare a cunostintelor teoretice si a abilitatilor practice.

### b. Principii de notare

Modul de calcul a notei: 2 p. teoria, 3 p. intrebarile; 4 p. problemele și 1 p. din oficiu.

### c. Informarea studentilor cu privire la evaluarea asociata disciplinei

In cadrul primului curs se face o prezentare generala a continutului cursului si a problemelor legate de maniera de evaluare a studentilor. Se vor preciza atat modalitatile de evaluare formativa (discutii si intrebari in cadrul fiecarui laborator) cat si cele de evaluare sumativa (discutii si intrebari in cadrul cursului dupa predarea unei parti bine delimitata din materia disciplinei).

**9. Responsabil de curs**

Nume : Sef lucrari dr. ing. Alexandru Stelian Stancu

Date de contact : Galati, str. Domneasca, nr. 111email- Cod postal: 800201 TEL/FAX: 0236/460182

Email : [Alexandru.Stancu@upe.edu](mailto:Alexandru.Stancu@upe.edu)

Responsabil de curs,



Sef Departament / Catedra,

