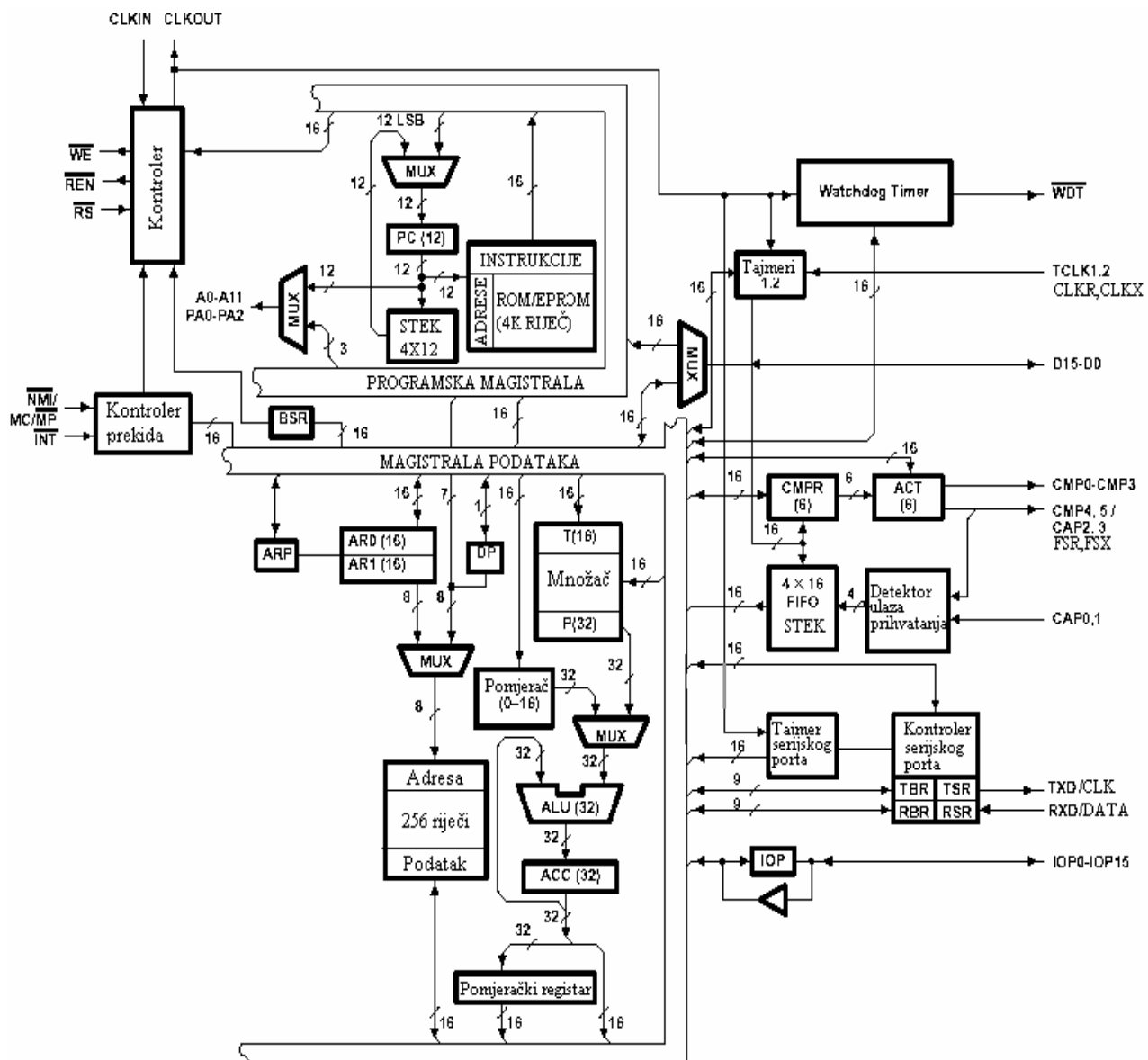


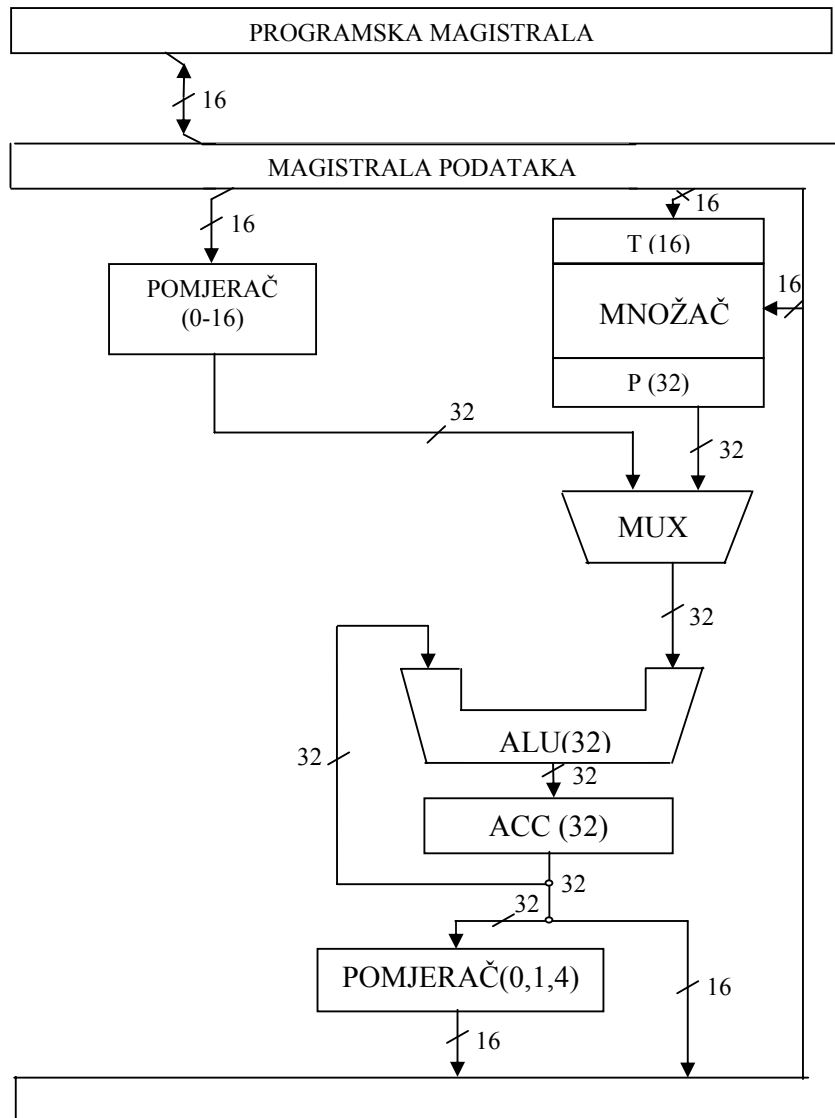
DSP TMS320C14 ПРОЦЕСОР

Карактеристике микропроцесора

1. Инструкциони циклус траје 160ns,
2. 256 ријечи (ријеч садржи 16 бита) интерне RAM меморије за податке,
3. 4k*16 интерне ROM или EPROM програмске меморије (укупно 4k*16 програмске меморије),
4. EPROM код – заштита против копирања,
5. 32b аакумулатор,
6. хардверски множач 16*16 који даје 32b резултат,
7. два 16b помјерачка регистра,
8. 7 улазних и 7 излазних канала,
9. 16b магистрала података чији је пропусни опсег 50 Mb/s,
10. 16b бит – адресибилни U-I порт,
11. серијски порт са програмибилним протоколима,
12. 4 независна тајмера (1 watchdog тајмер, 2 опште намјене и 1 за одређивање брзине серијског преноса),
13. контролер догађаја у којем се налазе двије јединице: систем за прихватање и систем за поређење,
14. 15 вањских и унутрашњих прекида,
15. 68 – пинско кућиште.



Функционални блок дијаграм процесора TMS320C14/E14



Централна аритметичко-логичка јединица (CALU)

Процесни хардвер

TMS320C14 садржи А-Л јединицу и акумулатор за подршку 32b аритметике у другом комплементу.

АЛУ је аритметичка јединица опште намјене која користи 16b ријечи узете из RAM меморије за податке, непосредни податак у саставу инструкције или 32b резултат множења похрањен у Р регистру за извођење А-Л операција.

У акумулатор се похрањује резултат А-Л операције. Такође, акумулатор садржи податак који се користи као улазни за АЛУ. Акумулатор је 32b и подијељен је на двије 16b ријечи, ријеч ниже тежине АСCL (0 – 15) и ријеч више тежине АСЧН (16 – 31).

Множач изводи 16*16 множење и даје 32b резултат у једном инструкционом циклусу. Множач се састоји од Т регистра, Р регистра и множачке матрице. У Т регистар се привремено смјешта 16b фактор, а у Р регистар похрањује се резултат множења. Други

фактор може бити податак из меморије или податак узет из инструкционе ријечи. Брзи (интерни) множачи омогућавају ефикасно извођење математичких интензивних алгоритама, као што су PID петље, Калманово филтрирање, дигитална обрада сигнала и слично.

Постоје два помјерачка регистра расположива за манипулацију подацима:

1. помјерачки регистар “barrel” изводи помјерање улијево податка из меморије у опсегу 0–16 бита.

2. помјерачки регистар помјера улијево 0, 1 или 4 мјеста целокупан садржај акумулатора и виша ријеч акумулатора из овог регистра просљеђује се на магистралу података.

Овај микропроцесор посједује STEK са 4 нивоа који се првенствено користи у раду са потпрограмима и прекидним рутинама.

Аритметичко-логичке операције

Извршење типичне аритметичко-логичке операције има сљедеће фазе:

1. податак се прихвата из RAM меморије на магистралу података,
2. податак пролази кроз помјерачки регистар, гдје може бити помјерен од 0 до 16 бита у зависности од вриједности специфициране у оквиру инструкције,
3. подаци улазе у аритметичко-логичку јединицу, обрађују се и резултат се пуни у акумулатор
4. резултат из акумулатора пролази кроз други помјерачки регистар у којем се скалира ради добијања коректног резултата,
5. резултат се меморише у RAM меморију за податке.

Код аритметичко-логичких операција један податак (операнд) је увијек из акумулатора, а други из Р регистра множача или из RAM меморије за податке (из RAM-а се помјера у помјерачком регистру у 2. кораку).

Периферијске јединице

U – I порт. TMS320C14 има 16-битни бит-адресибилни U-I порт који има један додјељени прекид IOINT

Тајмери. Овај процесор има 4 тајмера (1 watchdog тајмер, 2 опште намјене и генератор брзине серијског преноса).

Функција watchdog тајмера је да спријечи софтверске блокаде.

Тајмери опште намјене могу да броје импулсе из интерног осцилатора или импулсе из вањског такт извора.

Генератор брзине серијског преноса користи се за генерисање брзине серијског преноса у серијској комуникацији.

Сваком тајмеру је додјељен прекид.

Контролер догађаја чине систем поређења CMP и систем прихватања CAP.

Систем поређења CMP омогућује поређење садржаја регистра са садржајем тајмерског регистра и у случају да су ове вриједности једнаке генерише се одговарајуће стање на вањским прикључцима (пиновима) придруженим систему поређења. Обично се користи за генерисање PWM управљачких сигнала (Pulse Width Modulation – импулсно ширинска модулација).

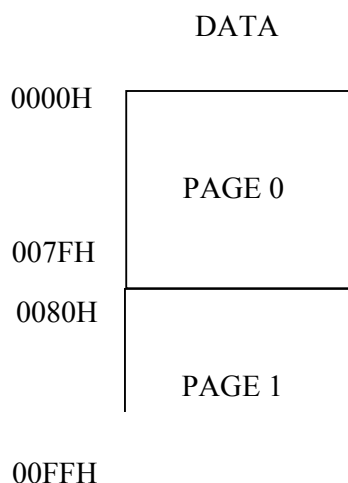
Систем поређења CAP омогућује регистровање импулса на улазу система, као и мјерење временског интервала појављивања ових импулса.

Серијски порт има 3 мода рада: синхрони, асинхрони и CODEC (кодек). Поддржава два протокола: протокол који се заснива на одређивању адресе и протокол који се заснива на изједначавању адресе. Има два додјељена прекида.

Организација меморије

TMS320C14(E14) има модификовану харвардску архитектуру са два одвојена простора за програме и податке и одвојеном интерном магистралом за податке и инструкције. Интерна или екстерна програмска меморија може се придружити CPU у зависности од одабраног мода рада (у зависности од бита MC/MP)

Меморија за податке. Микропроцесор садржи 256*16 RAM простора за меморисање података. RAM меморија за податке је организована на сљедећи начин:



Програмска меморија. Поред унутрашње програмске меморије 4k*16 бита ROM-а или EPROM-а микропроцесор може приступити и вањској меморији. Селекција меморијског ресурса врши се битом MC/MP у SYSCON регистру постављањем овог бита.



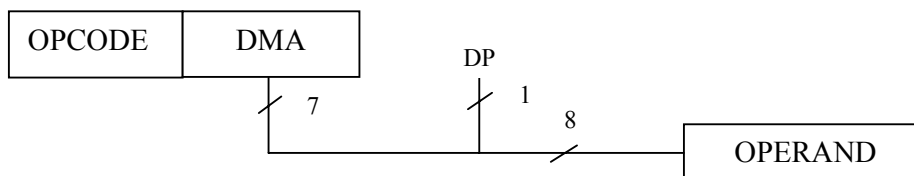
Ова могућност промјене MC/MP бита даје механизам за јединствено проширење програмске меморије са 4k на максималних 8k ријечи. Највиших 8 ријечи вањске меморије резервисани су за вањске U-I портове. TMS320E14 има и заштитни бит који када је постављен онемогућује читање EPROMA. 96 ријечи у ROM меморији код TMS320C14 резервисани су за интерну употребу од стране TXSAS (INSASE).

Помоћни регистри. Микропроцесор TMS320C14(E14) има два 16-битна помоћна регистра AR0 и AR1. Ови регистри се могу користити за неколико функција: за индивидуално адресирање меморије за податке, привремено меморисање података и реализацију бројчаних петљи, индиректно смјештање меморијске адресе операнда у 8 најнижих бита помоћног регистра. Регистри се селекују једним битом ARP који се поставља на 0 и 1. Ако је ARP=1 селекуван је AR1. ARP је дио статусног регистра који може бити меморисан у меморији.

Адресирање меморије

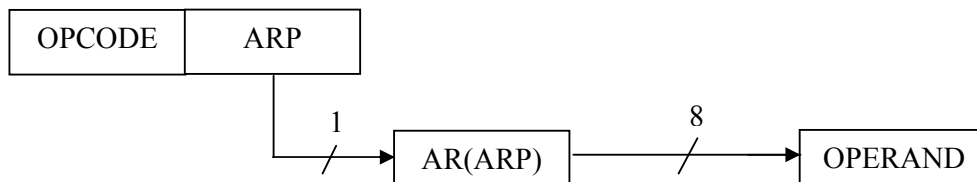
Код микропроцесора TMS320C14(E14) може се адресирати до 4k ријечи у програмској меморији и до 256 ријечи у меморији за податке. Могу се користити три начина адресирања операнда:

1. Директно адресирање:



Код директног адресирања 1 бит DP (показивач меморијске странице) селекује страницу 0 (DP0) или страницу 1 (DP1). Меморијска адреса специфицирана је са 7 нижих бита инструкције који заједно са DP адресирају жељену ријеч унутар странице меморије за податке.

2. Индиректно адресирање:



Индиректно адресирање користи 8 нижих бита помоћних регистара као меморијску адресу.

3. Непосредно адресирање:



Множење

TMS320C14(E14) хардверски множач изводи (16 бита) x (16 бита) множење у комплемент 2 аритметици и даје 32 – битни резултат у једном машинском циклусу. Код множења један операнд је увијек у Т регистру, док се други прибавља у множач у току саме инструкције множења. Резултат множења смјешта се у Р регистар. Прије слjedeће операције множења претходни резултат мора бити пребачен из Р регистра акумулатора, како не би био изгубљен. Израчунавања на TMS320C14(E14) процесору заснована су на представљању бројева у фиксном зарезу и комплемент 2 аритметици. Сваки 16 битни број представљен је са битом знака, i цијелобројних битова и $15-i$ битова иза бинарног зареза.

Број 0 000010 1010 0000 има вриједност 2.625
| - бинарна тачка

За овај број кажемо да је представљен у Q_8 формату. Опсег овог формата је између -128 (1000 000 000 000) и 127.996 (0111 1111 1111 1111). Тачност представљања бројева у Q_8 формату је око $0.004 \left(\frac{1}{2^8}, \text{односно } \frac{1}{256} \right)$.

У Q формату број уз Q (15 у Q_{15}) означава колико бита се налази десно од бинарне тачке. Бројеви представљени у Q_{15} могу узети вриједности само у опсегу +1 (представљамо са 0.99997...) до -1.

$$\begin{array}{r} 0100\ 0000\ 0000\ 0000 = 0.5 \text{ у } Q_{15} \\ \times 0100\ 0000\ 0000\ 0000 = 0.5 \text{ у } Q_{15} \\ \hline \end{array}$$

00 01 0000 0000 0000 0000 0000 0000 = 0.25 у Q_{15}
| - бинарна тачка

Након множења бројева представљених у Q_{15} формату добијају се два бита знака. Често, прецизност од 16 бита је практичнија него чување свих 32 бита. Међутим виших 16 бита резултата не садрже свих 15 бита резултата у Q_{15} формату, јер операција множења креира други бит знака. За коректан резултат потребно је производ помјерити за један бит улијево, као што је приказано у слjedeћем примјеру:

$$\begin{array}{l} \text{LT} \quad \text{OP1} \quad ; \quad \text{OP1} = 4000_{\text{H}} \quad (0.5 \text{ у } Q_{15}) \\ \text{MPY} \quad \text{OP2} \quad ; \quad \text{OP2} = 4000_{\text{H}} \quad (0.5 \text{ у } Q_{15}) \\ \text{PAC} \\ \text{SACH} \quad \text{ANS},1 \quad ; \quad \text{ANS} = 2000_{\text{H}} \quad (0.25 \text{ у } Q_{15}) \end{array}$$

MPYK инструкција изводи множење са 13 – битним означеним константама, што значи да се Q_{15} број може помножити са Q_{12} бројем. Резултат мора бити помјерен 4 бита улијево да би се добио Q_{15} резултат у пуној прецизности.

Примјер:

$$\begin{array}{l} \text{LT} \quad \text{OP1} \quad ; \quad \text{OP1} = 4000_{\text{H}} \quad (0.5 \text{ у } Q_{15}) \\ \text{MPYK} \quad 2048 \quad ; \quad \text{OP2} = 0800_{\text{H}} \quad (0.5 \text{ у } Q_{12}) \\ \text{PAC} \\ \text{SACH} \quad \text{ANS},4 \quad ; \quad \text{ANS} = 2000_{\text{H}} \quad (0.25 \text{ у } Q_{15}) \end{array}$$

Integer x Integer:

$$\begin{array}{r} 0000\ 0000\ 0001\ 0001 = 17\ \text{у}\ Q_0 \\ \times\ 1111\ 1111\ 1111\ 1011 = -5\ \text{у}\ Q_0 \\ \hline \end{array}$$

$$\begin{array}{r} 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1010\ 1011 = -85\ \text{у}\ Q_0 \\ | \text{- бинарна тачка} \end{array}$$

У овом случају посебан бит знака не мијења резултат, а жељени производ цио се налази у 16 нижих битова производа, као што је приказано у сљедећем примјеру:

LT OP1 ; OP1 = 0011_H (17 у Q₀)
MPY OP2 ; OP2 = FFFB_H (-5 у Q₀)
PAC
SACL ANS ; ANS = FFAB_H (-85 у Q₀)

Нотација Q₁₄ x Q₁₄

$$\begin{array}{r} 0110\ 0000\ 0000\ 0000 = 1.50\ \text{у}\ Q_{14} \\ \times\ 0011\ 0000\ 0000\ 0000 = 0.75\ \text{у}\ Q_{14} \\ \hline \end{array}$$

$$0001\ 0010\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000 = 1.125\ \text{у}\ Q_{28}$$

Максимална вриједност у Q₁₄ формату је 2, тако да је максимална вриједност производа 4. Два бита се користе за представљање цјелобројног дијела, а максимална тачност производа је 13 битова. Генерално, примјењује се сљедеће правило: производ броја са i цјелобројних битова и f битова иза бинарне тачке и броја са j цјелобројних битова и g битова иза бинарне тачке је број са $i+j$ цјелобројних битова и $f+g$ битова иза бинарне тачке. Највећа могућа прецизност за 16 – битну презентацију овог производа имаће $i+j$ цјелобројних битова и $15-i-j$ битова иза бинарне тачке.

Примјер:

LT OP1 ; OP1 = 6000_H (1.5 у Q₁₄)
MPY OP2 ; OP2 = 3000_H (0.75 у Q₁₄)
PAC
SACH ANS,1 ; ANS = 2400_H (1.125 у Q₁₃)

DSP алгоритми. Честа операција у DSP алгоритмима је сабирање производа. Садржај Р регистра додаје се акумулатору, а двије вриједности симултано се читају и множе. Једна вриједност из Р регистра множи се са двије вриједности из меморије за податке.

Примјер:

Имплементација множења и сабирања кориштењем LTA-MPY инструкционог пара.

ZAC		; $0 \rightarrow ACC$
LT	D1	; $D1 \rightarrow T$
MPY	C1	; $D1 * C1 \rightarrow P$
LTA	D2	; $D2 \rightarrow T \quad 0 + D1 * C1 \rightarrow ACC$
MPY	C2	; $D2 * C2 \rightarrow P$
LTA	D3	; $D3 \rightarrow T \quad D1 * C1 + D2 * C2 \rightarrow ACC$
	.	
	.	
	.	
LTA	DN	; $DN \rightarrow T \quad \sum_{i=1}^{n-1} D_i * C_i \rightarrow ACC$
MPY	CN	; $DN * CN$
APAC		; $\sum_{i=1}^{n-1} D_i * N_i + D_n * C_n \rightarrow ACC$

Дијелeње

Бинарно дијелeње је инверзна операција у односу на бинарно множење. Множење се састоји од више циклуса операција сабирања и помјерања, док се дијелeње може реализовати кроз више циклуса одузимања и помјерања. TMS320C14(E14) нема инструкцију за дијелeње, али се оно може ефикасно имплементирати кориштењем инструкције условног одузимања SUBC. За дати 16 – битни позитивни дјеленик и 16 – битни позитивни дјелилац, понављањем SUBC инструкције 16 пута добија 16 – битни количник у нижих 16 битова акумулатора и остатак у виших 16 битова акумулатора. У свакој SUBC инструкцији дјелилац помјерен за 15 бита улијево одузима се од акумулатора (или дјеленик). За свако одузимање које даје ненегативан резултат, резултат се смјешта у акумулатор, помјера за једно мјесто и најнижи бит акумулатора поставља на 1. За свако одузимање које даје негативан резултат, садржај акумулатора се помјера за једно мјесто улијево. Помјерањем остатка и количника након сваког одузимања добија се одвајање количника (у нижих 16) и остатка (у виших 16 бита акумулатора).

Једино ограничење за употребу SUBC инструкције је да и дјеленик и дјелилац морају бити позитивни. Због тога знак количника мора бити одређен, а количник израчунат кориштењем апсолутних вриједности дјеленика и дјелиоца. Када се имплементира алгоритам дијелeња важно је знати, ако се количник може представити као разломак, са одређеним бројем мјеста иза децималне тачке, који је степен тачности са којим количник може бити израчунат.

SUBC метод (примјер дијелeња броја 35 са 5):

32 виша ријеч ACC-а	нижа ријеч ACC-а 0
0000 0000 0000 0000	0000 0000 0010 0001
	-10 1000 0000 0000 0000
-10 0111 1111 1101 1111	

1° Дјеленик се учитава у ACC. Дјелилац се помјера 15 мјеста улијево и одузима се од садржаја ACC-а. Резултат одузимања је негативан, тако да се одбацује, а садржај ACC-а помјера један бит улијево.

0000 0000 0000 0000 0000

32 виша ријеч ACC-а	нижа ријеч ACC-а 0
0000 0000 0000 0000	0000 0000 0100 0010
	-10 1000 0000 0000 0000
-10 0111 1111 1011 1110	

2° Поновним одузимањем добија се негативан резултат, који се одбацује, а садржај ACC-а помјера се један бит улијево.

32 виша ријеч ACC-а	нижа ријеч ACC-а 0
0000 0000 0000 0100	0010 0000 0000 0010
	-10 1000 0000 0000 0000
0000 0000 0000 0001 1010 0000 0000 0000	

.....

14° Резултат је позитиван. Помјера се једно мјесто учијево и најнижи бит АСС поставља се на 1.

32 виша ријеч АСС-а	нижа ријеч АСС-а 0
0000 0000 0000 0011	0100 0000 0000 0001
-10	1000 0000 0000 0000
<hr/>	
0000 0000 0000 0000	1100 0000 0000 0001

Задаци

Задатак 1: Написати програмску рутину за дијелење два бинарна броја представљена у комплемент два формату произвољног знака. Количник треба да има специфицирану тачност. Тачност дијелења одређена је бројем цифара иза децималне тачке и меморисана је у меморијској локацији FRAC. Дјеленик се налази у меморијској локацији чија је симболичка адреса NUMERA, а дјелитељ у меморијској локацији DENOM. Количник треба смјестити у меморијску локацију QUOT. За привремено меморисање знака резултата програмеру је на располагању меморијска локација TEMSEN.

```
DN1:  LT    NUMERA
      MPY   DENOM
      PAC
      SACH  TEMSEN
      LAC   DENOM
      ABS
      SACL  DENOM
      LACK  15
      ADD   FRAC
      SACL  FRAC
      LAC   NUMERA
      ABS
      LAR   0,FRAC
LOOP: SUBC  DENOM
      BANZ  LOOP
      SACL  QUOT
      LAC   TEMSEN
      BGEZ  DONE
      ZAC
      SUB   QUAT
      SACL  QUAT
DONE: RET
```

Задатак 2: Написати програмску рутину за имплементацију PID алгоритма.

$$u(t) = K_p e(t) + K_I \int e(t) dt + K_d \frac{de(t)}{dt}$$

$u(t)$ - управљачко дјеловање

$e(t)$ - сигнал грешке

$$u(n) = u(n-1) + K_0 e(n) + K_1 e(n-1) + K_2 e(n-2)$$

```

.data                ;segment podataka
.bss    UN,1        ;izlaz kontrolera
.bss    E0,1        ;posljednji odmjerak signala greske
.bss    E1,1        ;pretposljednji odmjerak signala greske
.bss    E2,1        ;najstariji odmjerak signala greske
.bss    K0,1
.bss    K1,1
.bss    K2,1
.text

PID: IN    E0,PA0    ;ucitavanje novog signala greske
      LAC  UN        ;u(n-1) -> ACC
      LT   E2        ;e(n-2) -> T
      MPY  K2        ;K2*e(n-2) -> T
      LTD  E1        ;e(n-1) -> T; e(n-1) -> e(n-2); n(n-1) + K2 * e(n-2) -> ACC
      MPY  K1        ;K1*e(N-2) -> P
      LTD  E0        ;e(n) -> T; e(n) -> e(n-1)
                        ;n(n-1) + K2 * e(n-2) + K1 * e(n-1) -> ACC
      MPY  K0        ;K0*e(n) -> P
      APAC                        ;(ACC)+(P) -> (ACC)
      SACH UN,1
      OUT  UN,PA1
      RET

```

Задатак 3: Сигнал струје мотора одмјерава се помоћу биполарног (+/- 10 V) 10-битног А/D конвертора. Старт конверзије обезбјеђује се уписом вриједности #START_VAL на доњих 10 бита садржаја прочитаног са адресе #READ_ADC, при чему је вриједност прочитаних горњих 6 бита неизвјесна. Потребно је израчунати квадрат струје и приказати добијени резултат као аналогни униполарни сигнал (0 – 10 V) на 10 – битном D/A конвертору чија је адреса #DAC_ADD и чији је MSB повезан са MSB сабирнице података. Максималном опсегу улазног сигнала мора одговарати максимални опсег излазног сигнала. Начинити програм за читавање А/D, рачун и упис на D/A асемблеру за микроконтролер по жељи.

Рјешење: Изабрани микроконтролер је TMS320C14. Усвојимо да се у локацији BUSY налази информација да ли је А/D конверзија у току и ако је завршена вриједност у меморијској локацији је 0.

```

.data

.bss BUSY,1
.bss TMP,1
.bss TMP1,1
.bss ONE,1
.bss TRI,1

TABLE
.word START_VAL
.word START_ADD
.word READ_ADD
.word DAC_ADD
.text

LACK 1
SACL ONE
LT ONE
MPYK TABLE
PAC
SACL POINT
TBLR TMP           ; Očitavanje vrijednosti START_VAL iz tabele
ADD ONE
SACL POINT
TBLR TMP1          ; Očitavanje vrijednosti START_ADD iz tabele
LAR 0,TMP1
LAC TMP
SACL*              ; Start A/D konverzije
L1 LAC BUSY
BGZ L1             ; Provjera da li je A/D konverzija završena
LAC POINT
ADD ONE
SACL POINT
TBLR TMP1
LAR 0,TMP1
LAC*               ; Očitavanje podatka nakon završetka A/D konverzije (X)
SACL TMP
LACK OFFH
SACL TMP1
LACK 3

```

SACL TRI	
LAC TRI,8	
ADD TMP1	; Dobijanje vrijednosti 03FFH kao $3 \cdot 2^8 + \text{FFH}$
AND TMP	
	; Dobijanje korektne 10-bitne vrijednosti
	; ((operand (10-bitni)) AND (03FFH))
SACL TMP	
LT TMP	
MPY TMP	; $X \cdot X = X^2$
LAC POINT	
ADD ONE	
SACL POINT	
TBLR TMP	
LAR 0,TMP	; Adresiranje D/A konvertora
SACH* ,13	
	; Slanje podatka X^2 na D/A konvertor. Vršiti se pomjeranje
	; sadržaja akumulatora za 13 mjesta kako bi se u 10 najviših
	; bita ACC-a dobila vrijednost koja se šalje na D/A konvertor
END	

СЕТ ИНСТРУКЦИЈА ЗА DSP TMS320C14 ПРОЦЕСОР

ТАБЕЛА 1. СИМБОЛИ ИНСТРУКЦИЈА

СИМБОЛ	ЗНАЧЕЊЕ
ACC	АКУМУЛАТОР
D	ПОДРУЧЈЕ МЕМОРИЈЕ АДРЕСЕ ПОДАТАКА
I	БИТ АДРЕСНОГ МОДА
K	ПОЉЕ НЕПОСРЕДНОГ ОПЕРАНДА
PA	3-БИТНО АДРЕСНО ПОЉЕ ПОРТА
R	1-БИТНО ПОЉЕ СПЕЦИФИКАЦИЈЕ ОПЕРАНДА ПОМОЋНОГ РЕГИСТРА
S	4-БИТНИ КОД ЛИЈЕВОГ ПОМЈЕРАЈА
X	3-БИТНО ПОЉЕ ЛИЈЕВОГ ПОМЈЕРАЈА АКУМУЛАТОРА

ТАБЕЛА 2. TMS320C14 СЕТ ИНСТРУКЦИЈА

ИНСТРУКЦИЈЕ ЗА РАД СА АКУМУЛАТОРОМ

МНЕМОНИК	ОПИС ИНСТРУКЦИЈЕ	БР.ЦИКЛ.	БР.РИЈЕЧИ
ABS	АПСОЛУТНА ВРИЈЕДНОСТ АКУМУЛАТОРА	1	1
ADD	ДОДАТИ АКУМУЛАТОРУ СА ПОМЈЕРАЊЕМ	1	1
ADDH	ДОДАТИ ВИШИМ БИТОВИМА АКУМУЛАТОРА	1	1
ADDS	ДОДАТИ АКУМУЛ. БЕЗ ЗНАКА ЕКСТЕНЗИЈЕ	1	1
AND	ЛОГИЧКО "И" СА АКУМУЛАТОРОМ	1	1
LAC	НАПУНИТИ АКУМ. СА ПОМЈЕРАЊЕМ	1	1
LACK	НАПУНИ АКУМ. НЕПОСРЕДНО	1	1
OR	ЛОГИЧКО "ИЛИ" СА АКУМ.	1	1
SACH	ПОХРАНИТИ ВИШЕ БИТЕ АКУМ. СА ПОМЈ.	1	1
SACL	ПОХРАНИТИ НИЖЕ БИТЕ АКУМ.	1	1
SUB	ОДУЗИМАЊЕ ОД АКУМ. СА ПОМЈЕРАЊЕМ	1	1
SUBC	УСЛОВНО ОДУЗИМАЊЕ (ЗА ДИЈЕЉЕЊЕ)	1	1
SUBH	ОДУЗИМАЊЕ ОД ВИШИХ БИТА АКУМ.	1	1
SUBS	ОДУЗИМАЊЕ ОД АКУМ. БЕЗ ЗНАКА ЕКСТЕНЗИЈЕ	1	1
XOR	ЛОГИЧКО "ЕЛИ" СА АКУМУЛАТОРОМ	1	1
ZAC	УПИСАТИ НУЛУ У АКУМУЛАТОР	1	1
ZALH	УПИСАТИ 0 И НАПУНИТИ ВИШЕ БИТЕ АКУМ.	1	1
ZALS	УПИСАТИ 0 И НАПУНИТИ НИЖЕ БИТЕ АКУМ. БЕЗ ЗНАКА ЕКСТЕНЗИЈЕ	1	1

ИНСТРУКЦИЈЕ ЗА РАД СА ПОМОЋНИМ РЕГИСТРИМА И ПОКАЗИВАЧИМА НА СТРАНИЦЕ ПОДАТАКА

МНЕМОНИК	ОПИС ИНСТРУКЦИЈЕ	БР.ЦИКЛ.	БР.РИЈЕЧИ
LAR	НАПУНИТИ ПОМОЋНИ РЕГИСТАР	1	1
LARK	НАПУНИТИ ПОМОЋНИ РЕГИСТАР НЕПОСРЕДНО	1	1
LARP	НАПУНИТИ ПОКАЗИВАЧ ПОМ. РЕГИСТРА НЕПОСРЕДНО	1	1
LDP	НАПУНИТИ ПОКАЗИВАЧ СТРАНИЦЕ МЕМОРИЈЕ ПОДАТАКА	1	1
LDPK	НАПУНИТИ ПОКАЗИВАЧ СТРАНИЦЕ МЕМОРИЈЕ ПОДАТАКА НЕПОСРЕДНО	1	1
MAR	ИЗМЈЕНИТИ ПОМОЋНИ РЕГИСТАР И ПОКАЗИВАЧ	1	1
SAR	ПОХРАНИТИ ПОМОЋНИ РЕГИСТАР	1	1

ИНСТРУКЦИЈЕ ГРАНАЊА

МНЕМОНИК	ОПИС ИНСТРУКЦИЈЕ	БР.ЦИКЛ.	БР.РИЈЕЧИ
B	БЕЗУСЛОВНО ГРАНАЊЕ	2	2
BANZ	ГРАНАЊЕ АКО ЈЕ ПОМ. РЕГИСТАР РАЗЛ. ОД 0	2	2
BGEZ	ГРАНАЊЕ АКО ЈЕ АКУМУЛАТОР \geq 0	2	2
BGZ	ГРАНАЊЕ АКО ЈЕ АКУМУЛАТОР $>$ 0	2	2
BLEZ	ГРАНАЊЕ АКО ЈЕ АКУМУЛАТОР \leq 0	2	2
BLZ	ГРАНАЊЕ АКО ЈЕ АКУМУЛАТОР $<$ 0	2	2
BNZ	ГРАНАЊЕ АКО ЈЕ АКУМУЛАТОР \neq 0	2	2
BV	ГРАНАЊЕ НА ПРЕКОРАЧЕЊЕ	2	2
BZ	ГРАНАЊЕ АКО ЈЕ АКУМУЛАТОР=0	2	2
CALA	ПОЗИВ ПОТПРОГРАМА ИЗ АКУМУЛАТОРА	2	1
CALL	ПОЗИВ ПОТПРОГРАМА НЕПОСРЕДНО	2	2
RET	ПОВРАТАК ИЗ ПОТПРОГРАМА ИЛИ ПРЕКИДНЕ РУТИНЕ	2	1

ИНСТРУКЦИЈЕ ЗА РАД СА Р РЕГИСТРОМ, Т РЕГИСТРОМ И МНОЖАЧЕМ

МНЕМОНИК	ОПИС ИНСТРУКЦИЈЕ	БР.ЦИКЛ.	БР.РИЈЕЧИ
ARAC	ДОДАТИ Р РЕГИСТАР НА АКУМУЛАТОР	1	1
LT	НАПУНИТИ Т РЕГИСТАР	1	1
LTA	LTA ОБЈЕДИЊУЈЕ ИНСТРУКЦИЈЕ LT И ARAC	1	1
LTD	ОБЈЕДИЊУЈЕ ИНСТРУКЦИЈЕ LT, ARAC И DMOV	1	1
MPY	МНОЖЕЊЕ СА Т РЕГИСТРОМ, РЕЗУЛТАТ У Р РЕГ.	1	1
MPYK	МНОЖЕЊЕ Т РЕГ. СА НЕПОСРЕДНИМ ОПЕРАНДОМ, РЕЗУЛТАТ У Р РЕГИСТАР	1	1
PAC	НАПУНИТИ АКУМУЛАТОР ИЗ Р РЕГИСТРА	1	1
SPAC	ОДУЗЕТИ Р РЕГИСТАР ОД АКУМУЛАТОРА	1	1

КОНТРОЛНЕ ИНСТРУКЦИЈЕ

МНЕМОНИК	ОПИС ИНСТРУКЦИЈЕ	БР.ЦИКЛ.	БР.РИЈЕЧИ
DINT	ОНЕМОГУЋЕЊЕ ПРЕКИДА	1	1
EINT	ОМОГУЋЕЊЕ ПРЕКИДА	1	1
LST	НАПУНИТИ СТАТУСНИ РЕГИСТАР	1	1
NOP	БЕЗ ОПЕРАЦИЈЕ	1	1
POP	POP СТЕК У АКУМУЛАТОР	2	1
PUSH	PUSH СТЕК ИЗ АКУМУЛАТОРА	2	1
ROVM	РЕСЕТ МОДА ПРЕКОРАЧЕЊА	1	1
SOVM	ПОСТАВЉАЊЕ МОДА ПРЕКОРАЧЕЊА	1	1
SST	ПОХРАЊИВАЊЕ СТАТУСНОГ РЕГИСТРА	1	1

ОПЕРАЦИЈЕ ЗА У/И И МЕМОРИЈУ ПОДАТАКА

МНЕМОНИК	ОПИС ИНСТРУКЦИЈЕ	БР.ЦИКЛ.	БР.РИЈЕЧИ
DMOV	КОПИРАТИ САДРЖАЈ ЛОКАЦИЈЕ МЕМОРИЈЕ ЗА ПОДАТКЕ У СЉЕДЕЋУ БИШУ ЛОКАЦИЈУ	1	1
IN	УЛАЗНИ ПОРТ ЗА ПОДАТКЕ	2	1
OUT	ИЗЛАЗНИ ПОРТ ЗА ПОДАТКЕ	2	
TBLR	ЧИТАЊЕ ТАБЕЛЕ ИЗ ПРОГРАМСКЕ МЕМОРИЈЕ У РАМ МЕМОРИЈУ ЗА ПОДАТКЕ	3	1
TBLW	УПИСИВАЊЕ ТАБЕЛЕ ИЗ RAM-а У ПРОГРАМСКУ МЕМОРИЈУ	3	1

САДРЖАЈ

➤ Карактеристике микропроцесора.....	1
➤ Функционални блок дијаграм.....	2
➤ Процесни хардвер.....	3
➤ Аритметичко-логичке операције.....	4
➤ Периферијске јединице.....	4
➤ Организација меморије.....	5
➤ Адресирање меморије.....	6
➤ Множење.....	7
➤ Дијељење.....	10
➤ Задаци.....	12
➤ Сет инструкција.....	14
➤ Садржај.....	18