

Зертханалық жұмыс №2

Пән: Компьютер архитектурасы және операциялар келісімділігі

Тақырыбы: MARS MIPS симуляторында арифметика-логикалық есептер шығару

Мақсаты:

- Ассемблер тілінде арифметикалық амалдарды орындауды үйрету;
- Логикалық операциялардың (AND, OR, XOR, NOT) жұмысын түсіндіру;
- Регистрлерді қолдана отырып есептер шығаруды үйрету.

Қысқаша теория:

Бұл зертханалық жұмыстың мақсаты — студенттерге MIPS ассемблерінде арифметикалық және логикалық операциялар қалай орындалатынын тәжірибеде көрсету. MIPS архитектурасы RISC (Reduced Instruction Set Computer) принципіне негізделген, яғни командалар қарапайым құрылымда және біртекті форматта орындалады.

Ассемблерлік бағдарламалар регистрлер арқылы жұмыс істейді.

Регистрлер — процессор ішіндегі жылдам

жад ұяшықтары. Көбіне уақытша регистрлер (\$t0–\$t9) есептеулер үшін пайдаланылады; \$a0–\$a3 аргументтер

үшін, ал \$v0–\$v1 нәтижені қайтару үшін қолданылады. Бағдарламалық код екі негізгі бөлімге бөлінеді:

.data (деректер) және .text (нұсқаулар). Мысалы, мәтіндік жолды .data бөлімінде анықтап, .text бөлімінде оны шығару үшін syscall қолданамыз.

Арифметикалық командалар add, sub, mul, div регистрлер арасында орындалады. add \$t0, \$t1, \$t2 командасы

\$t0 = \$t1 + \$t2 орындаса, sub — азайту, mul — көбейту, div — бөлу

операциясын орындайды. Бөлудің нәтижесі және қалдығы кейбір жағдайларда арнайы регистрге (HI/LO) жазылатынына назар аудару керек, алайда MARS симуляторында бұл процесс жеңілдетілген түрде көрсетіледі. Арифметикалық есептерді орындаған кезде өрістердің шегін (overflow) және бүтіндік сипатын ескерген жөн.

Логикалық командалар (and, or, xor, nor) биттік деңгейде жұмыс жасайды. Олар екі регистрдің сәйкес биттерін салыстырып, жаңа биттік нәтижені құрайды. Логикалық амалдар маскалау, байттар мен өрістерді ауыстыру сияқты төменгі деңгейлі операциялар үшін өте пайдалы. Сонымен қатар shift командалары (sll, srl, sra) биттерді жылжытады және белгілі бір есептерде (2^n көбейтуді жеңілдету) тиімді қолданылады.

MARS-та енгізу/шығару syscall арқылы жүзеге асады: li \$v0, 1 — бүтін сан шығару, li \$v0, 4 — жол шығару, li \$v0, 5 — бүтін сан енгізу. Практикада студенттер жүйелік шақырулар арқылы енгізілген мәндерді алып, оларды регистрлерде өңдеп, нәтижені шығарады. Құжатта келтірілген мысалдар арифметикалық және логикалық амалдардың негізгі қолданылуын көрсетеді және студенттің регистрлерді басқару мен syscall-ды қолдану тәжірибесін қалыптастырады.

Мысалдар (код үлгілері және түсініктеме):

Мысал 1. Қосу операциясы (Addition)

.data

msg: .asciiz "Нәтиже: "

.text

main:

li \$t1, 15 # Бірінші санды t1-ге жүктеу

```
li $t2, 7      # Екінші санды t2-ге жүктеу  
add $t3, $t1, $t2 # t3 = t1 + t2 (қосу операциясы)
```

```
# Нәтижені шығару  
li $v0, 4      # жол шығару қызметі  
la $a0, msg  
syscall
```

```
li $v0, 1      # бүтін санды шығару  
move $a0, $t3  
syscall
```

```
li $v0, 10     # бағдарлама аяқтау  
syscall
```

Түсіндірме: \$t1 және \$t2 регистрлеріне екі мән жүктеліп, add командасы арқылы қосынды есептеледі. Қосынды \$t3 регистрінде сақталып, syscall арқылы шығарылады.

Мысал 2. Биттік AND (Bitwise AND)

```
.text
```

```
main:
```

```
li $t1, 0xF0F0 # екілік түрде 1111000011110000  
li $t2, 0x0FF0 # екілік түрде 0000111111110000  
and $t3, $t1, $t2 # биттік AND
```

```
li $v0, 1  
move $a0, $t3  
syscall
```

```
li $v0, 10  
syscall
```

Түсіндірме: and командасы әр бит бойынша AND операциясын орындайды. Бұл код мысалында \$t3 регистрінде AND нәтижесі сақталып, экранға шығарылады.

№2 зертханалық жұмыстың тапсырмалары.

1. Екі бүтін санды қосып, нәтижесін шығаратын бағдарлама жазыңыз. (пайдаланушы енгізуі арқылы ұйымдастырыңыз).
2. Екі санды азайтып, олардың абсолютті айырмасын шығаратын бағдарлама жасаңыз.
3. Екі сан үшін AND, OR, XOR операцияларын орындап, әр нәтиженің мәнін шығарыңыз.
4. Енгізілген бүтін санның n-ші битінің мәнін анықтайтын бағдарлама жазыңыз (shift операцияларын қолданыңыз).
5. Пайдаланушы енгізген N саны үшін 1-ден N-ге дейінгі сандардың қосындысын есептеп шығарыңыз (циклды пайдаланыңыз).