

Министерство образования и науки Российской
Федерации
ГОУ ВПО АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ
Физико-технический факультет
Кафедра вычислительной техники и электроники (ВТиЭ)

ТРАНСЛЯТОР ЯЗЫКА C В ЯЗЫК ПРОЦЕССОРА SEAFORTH40

Выполнил:
студент 555 гр. А.А. Лыгин
Научный руководитель:
к. ф.-м. н. ст. преп. каф. ВТиЭ А.В. Калачев

Барнаул 2010



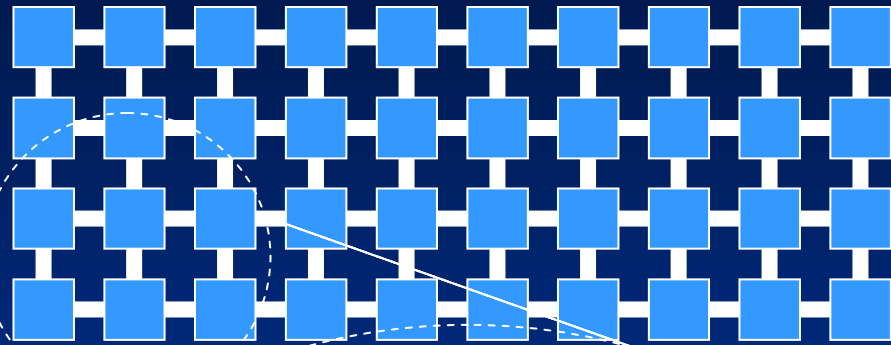
Цель работы – создание транслятора языка C в диалект языка Forth для процессора SEAforth40 - VentureForth

Задачи работы:

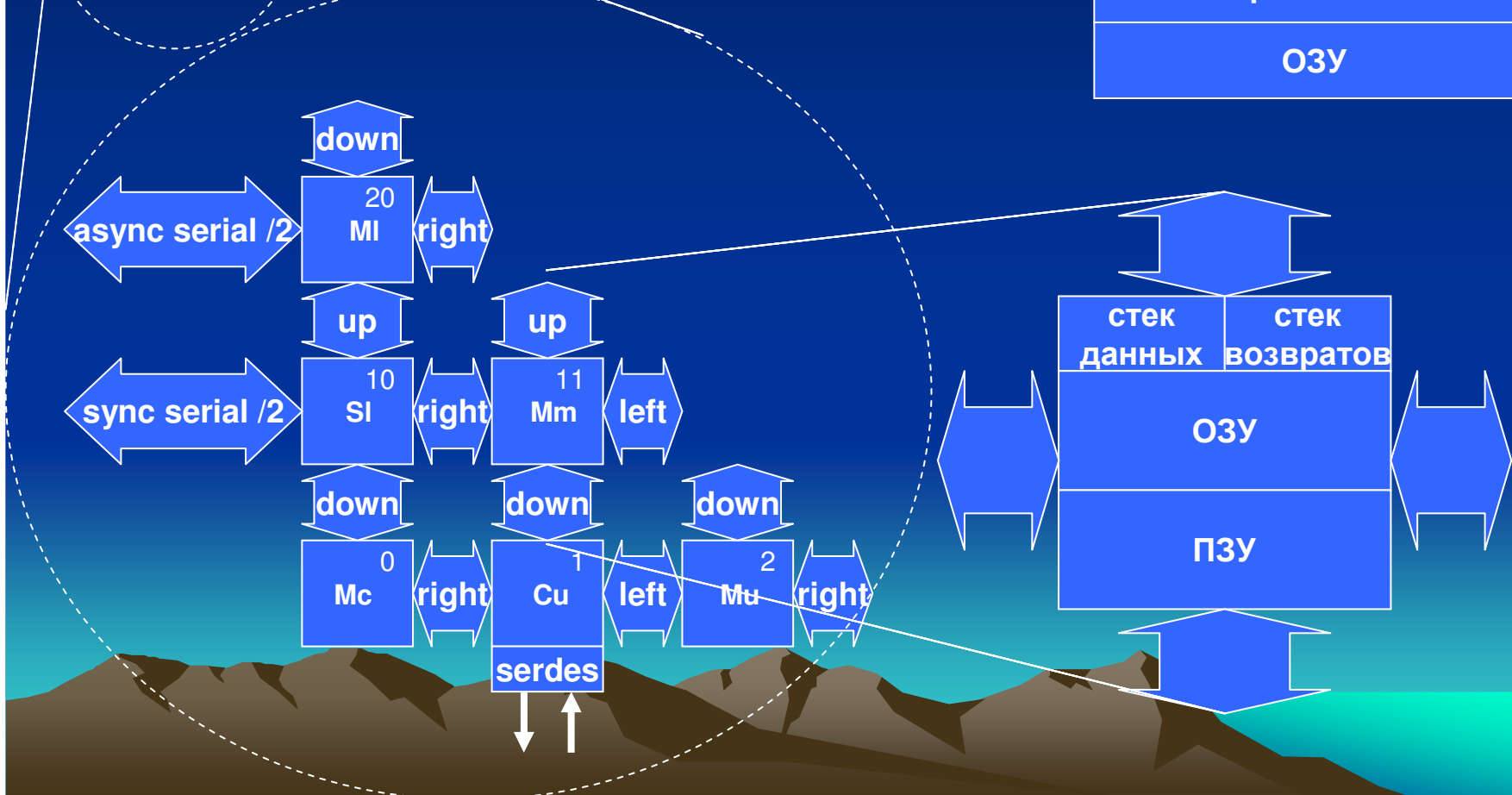
1. изучить специфику целевой платформы;
2. описать общий подход к трансляции на основе онтологии;
3. реализовать транслятор со счетчиком кода и созданием проектных файлов, совместимых с фирменной средой разработки.



Архитектурные особенности SEAforth40



Специальные регистры Коммуникационные порты	1FF
Зеркало ПЗУ	100 FF C0
ПЗУ	BF 80
Зеркало ОЗУ	7F 40
ОЗУ	3F 0



Система команд

call ; return jump ; : if -if next unext	Branch	Memory
!b !a !p+ !a+ @b @a @a+ @p+	Memory	

. (nop) push pop dup drop over	Stack	ALU
b! a! a@	Register	
and xor not	Bit-wise	
+ +* 2/ 2*	Math	

Формирование программного проекта

```
v.VF +include "c7Fr01/romconfig.f"
```

```
33 {node 0 org 'r--- =p node}  
10 {node 0 org here =p include toggle17-1.vf node}  
20 {node 0 org here =p include toggle17-1.vf node}  
29 {node 0 org here =p include blinkLED.vf node}  
36 {node 0 org here =p include sawtooth.vf node}  
37 {node 0 org here =p include sawtooth.vf node}  
39 {node 0 org here =p include sawtooth.vf node}
```

```
19 29 39 38 37 36 35 34 33 32 31 30 20 10 14 nodePath
```

```
reset
```

```
cr
```

```
[x] find-drive /USBdrive [x]>USBdrive close-drive
```

```
\ blinkLED.vf
```

```
$8000 1 >stk
```

```
: blinkLed
```

```
$3FE00 # a!
```

```
begin
```

```
620 # for
```

```
3 # !b
```

```
dup for
```

```
... unext \ delay
```

```
2 # !b
```

```
dup not for
```

```
... unext \ delay
```

```
next
```

```
$10000 # . + a@ and
```

```
| again
```

1. определяем узел
2. определяем адрес начала компиляции
3. инициализация программного счетчика
4. снабжаем узел кодом
5. активируем узел

```
10 {node  
0 org  
here =p  
include blinkLED.vf  
node}
```

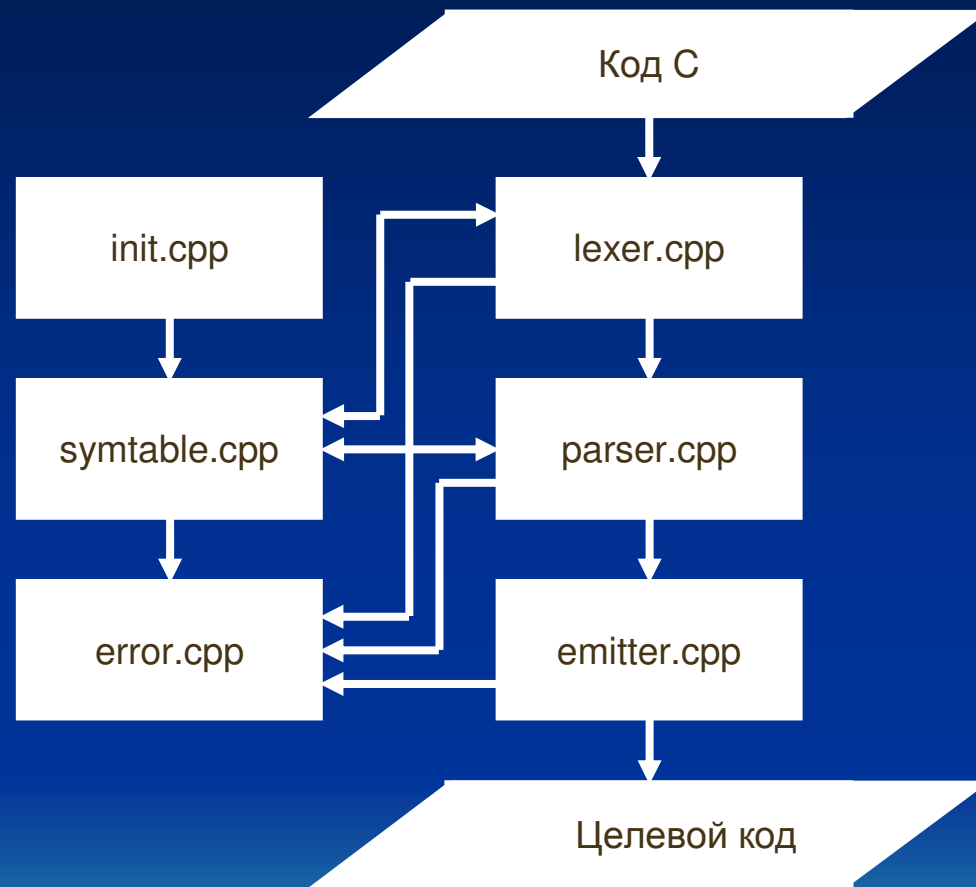
Приоритеты операций языка C

Приоритет	Оператор	Комментарий
1	()	делитель
2	!	логическое НЕ
2	++	унарный префиксный инкремент
2	--	унарный префиксный декремент
2	-	унарный префиксный оператор знака
2	+	унарный префиксный оператор знака
3	*	бинарный инфиксный оператор умножения
3	/	бинарный инфиксный оператор деления
3	%	бинарный инфиксный оператор остатка от деления
4	+	бинарный инфиксный оператор сложения
4	-	бинарный инфиксный оператор вычитания
5	<	бинарный инфиксный оператор порядка
5	<=	бинарный инфиксный оператор порядка
5	>	бинарный инфиксный оператор порядка
5	>=	бинарный инфиксный оператор порядка

Приоритеты операций языка C

Приоритет	Оператор	Комментарий
6	==	бинарный инфиксный оператор эквивалентности
6	!=	бинарный инфиксный оператор неэквивалентности
7	&	бинарный инфиксный битовый оператор И
8	^	бинарный инфиксный битовый оператор исключающего ИЛИ
9		бинарный инфиксный битовый оператор ИЛИ
10	*=	присваивание с операцией
10	/=	присваивание с операцией
10	%=	присваивание с операцией
10	+=	присваивание операцией
10	-=	присваивание операцией
10	&=	присваивание операцией
10	^=	присваивание операцией
10	=	присваивание операцией
10	=	присваивание
11	,	делитель

СХЕМА ВЗАИМОДЕЙСТВИЯ МОДУЛЕЙ



Сопоставление конструкций

Конструкция C	Конструкция VentureForth	Комментарий
<code>int ID;</code>	<code>here equ ID 0 ,</code>	объявление переменной
<code>ID=fgetc(STREAM)</code>	<code>STREAM b! @b ID b! !b</code>	ввод данных из потока в адресуемую ячейку
<code>fputc(EXPR,STREAM);</code>	<code>EXPR STREAM b! !b</code>	вывод данных в поток (потоки)
<code>ID=ID+EXPR</code>	<code>EXPR ID b! @b . + ID b! !b</code>	переопределение переменной
<code>ID+=EXPR;</code>	<code>EXPR ID b! @b . + !b</code>	переопределение переменной
<code>ID=ID-NUM</code>	<code>ID b! @b NUM negate . + ID b! !b</code>	переопределение переменной
<code>--ID</code>	<code>ID b! @b \$1FF # . + !b</code>	декремент переменной
<code>if(FLAG)STMT; else STMT;</code>	<code>FLAG if STMT else STMT then</code>	ветвление
<code>while(FLAG)STMT;</code>	<code>begin FLAG while STMT repeat</code>	«неопределенный цикл»

Пример трансляции

```
//node11.cpp
#include <cstdlib>
#include <iostream>

int main(){
    int buf,sum=0;
    buf=fgetc(R);
    int flag=true;
    while(buf>0&flag){
        if(sum>255)flag=false;
        else
            sum+=buf;
        buf=fgetc(R);
    }
    fputc(sum,L);
    return -67;
}
```

```
\ node11.vf
here equ buf 0 ,
here equ sum 0 ,
here equ flag 0 ,
: > not 1 # . + . + -if drop $1FF # ; then drop 0 # ;
: negate not 1 # . + ;

0 # dup sum # b! !b buf # b! !b
'r--- # b! @b buf # b! !b

$1FF # flag # b! !b
begin
buf # b! @b 0 # > flag # b! @b and
while
sum # b! @b 255 # > if
0 # flag # b! !b
else
buf # b! @b sum # b! @b . + !b
then

'r--- # b! @b buf # b! !b
repeat
sum # b! @b '-l- # b! !b
67 # negate
Total: 25 words.
```

Пример трансляции

```
//node12.cpp
#include <cstdlib>
#include <iostream>

int main(){
    int a=1;
    while(a<25){
        a+=2;
        a*=(-1);
        if(a>0)fputc(a,R);
        else
            fputc(a,L);
    }
    return -67;
}
```

```
v.VF +include "c7Fr01/romconfig.f"
```

```
11 {node $E org here =p include node11.vf node}
12 {node $A org here =p include node12.vf node}
13 {node $E org here =p include node13.vf node}
```

```
\ node12.vf
here equ a 0 ,
: < not 1 # . + . + -if drop $1FF # ; then drop 0 # ;
: negate not 1 # . + ;
: > not 1 # . + . + -if drop $1FF # ; then drop 0 # ;

1 # a # b! !b
begin
a # b! @b 25 # <
while
2 # a # b! @b . + !b
1 # negate
a # b! @b * !b
a # b! @b 0 # > if
a # b! @b 'r--- # b! !b
else
a # b! @b '--l- # b! !b
then

repeat
67 # negate
Total: 20 words.
```

Пример трансляции

```
//node13.cpp
#include <cstdlib>
#include <iostream>

int main(){
    int buf,sum=0;
    buf=fgetc(L);
    int flag=true;
    while(buf>0&flag){
        if(sum>255)flag=false;
        else
            sum+=buf;
        buf=fgetc(L);
    }
    fputc(sum,R);
return -67;
}
```

```
\ node13.vf
here equ buf 0 ,
here equ sum 0 ,
here equ flag 0 ,
: > not 1 # . + . + -if drop $1FF # ; then drop 0 # ;
: negate not 1 # . + ;

0 # dup sum # b! !b buf # b! !b
'--l- # b! @b buf # b! !b

$1FF # flag # b! !b
begin
buf # b! @b 0 # > flag # b! @b and
while
sum # b! @b 255 # > if
0 # flag # b! !b
else
buf # b! @b sum # b! @b . + !b
then

'--l- # b! @b buf # b! !b
repeat
sum # b! @b 'r--- # b! !b
67 # negate
Total: 25 words.
```

Заключение

В результате выполнения дипломного проекта была изучена технология многоядерных процессоров SEAForth, рассмотрены способы программирования на языке VentureForth. В необходимой мере обобщены классы (внутри одной предметной области) языков программирования C и VentureForth. Подробно описаны ключевые шаги разработки, обоснованы принятые решения.

Общий подход к разработке транслятора организован с помощью неформального построения онтологии предметной области (без четкого определения в теоретико-модельных или иных терминах).

Для более удобной работы предусмотрена автоматическая сборка проектных файлов. Для оценки необходимого объема оперативной памяти реализован счетчик команд, в конце работы приведено руководство пользователя.

