МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования
«Гомельский государственный университет
имени Франциска Скорины»

Факультет математики и технологий программирования

Кафедра математических проблем управления и информатики

# Развитие библиотек функций C-МПА для поддержки генерации HLCCAD-схем программами искусственного интеллекта

Курсовая работа

Исполнитель
студент группы ПО-21 _____ М.В.Малашков

Научный руководитель
доцент кафедры МПУИ,
к.т.н., доцент _____М.С.Долинский

Гомель 2024

# РЕФЕРАТ

Курсовая работа 30 страниц, 5 рисунков, 13 таблиц, 4 источника, 2 приложения.

Ключевые слова: C-МПА, HLCCAD, IA, схемы.

Цель работы: Развитие библиотек функций C-МПА для поддержки генерации HLCCAD-схем программами искусственного интеллекта.

Методология проведения работы: изучение работы с схемами HLCCAD, изучение имеющихся функций C-МПА, создание функций устройств HLCCAD на C-МПА, применение новых функций для решения задач.

Полученные результаты работы: разработаны функции для решения некоторых областей программирования с C-МПА.

Область применения результатов: программирование AI.

Практическая значимость работы: разработанные функции делают вклад в создание IA для помощи студентам в работе с HLCCAD.

Выводы: были созданы функции библиотек HLCCAD на языке C-МПА.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ . . . . . 4
1 Библиотеки Standard, SignedArithm, New . . . . . 5
.
2 Примеры задач решаемых с помощью данных библиотек . . . . . 1
3 Примеры задач с некорректным созданием С-МПА программы. . . . . 1
. . . .
4 Примеры задач, не решаемых только с помощью библиотек
Standard, SignedArithm, New . . . . . 1
. . . . . .
5 Библиотека EXTENDED. . . . . 1
6 Дальнейшие направления работы. . . . . 1
ЗАКЛЮЧЕНИЕ. . . . . 1
ПРИЛОЖЕНИЕ 1

# ВВЕДЕНИЕ

В современном динамичном мире, пронизанном технологическими инновациями, искусственный интеллект становится неотъемлемой частью нашей повседневной жизни, взаимодействуя с различными сферами деятельности и оказывая значительное воздействие на социальные, экономические и культурные аспекты. Все более усиливающийся вклад искусственного интеллекта в различные области, начиная от медицины и образования, и заканчивая промышленностью и исследованиями, подчеркивает важность изучения и понимания этой технологии.

Использование искусственного интеллекта (AI) в обучении студентов в сфере информационных технологий (IT) приносит существенные преимущества. Эта инновационная технология позволяет создавать персонализированные программы обучения, учитывая уровень знаний и особенности каждого студента. Такое индивидуализированное обучение способствует эффективному использованию времени и повышает качество обучения.

Дабы развивать новые технологии и не отставать от инноваций М.С. Долинским было предложено разработать IA для упрощения обучения новых студентов работе с HLCCAD. Изначально предполагалось обучать IA при помощи схем, созданных на базе HLCCAD, но в дальнейшем было выяснено, что AI проще обучать при помощи кода, поэтому началась разработка на языке C++. Обучение искусственного интеллекта продолжилось при помощи C-МПА, а именно в нём создавались функции, аналогичные устройствам HLCCAD, и использовались для решения задач. Изначально были разработаны функции библиотек Standard, SignedArithm, New, но в процессе выполнения курсовой работы я разработал новую библиотеку EXTENDED.

В данной курсовой работе я помогаю нашей цели: создать IA для помощи новым студентам в освоении HLCCAD, путём расширения библиотек функций на C-МПА, аналогичных устройствам HLCCAD. До моей курсовой работы были созданы и другие функции для работы с IA, однако в ходе реализации некоторых задач выяснилось, что уже созданных функций и процедур было недостаточно для преобразования комплекса схем в C-МПА программы.

Далее в ходе ознакомления с самой работой будут представлены корректные и некорректные решения задач, а также объяснения принципов работы старых и новых функций HLCCAD. Мною будут описаны дальнейшие направления развития IA.

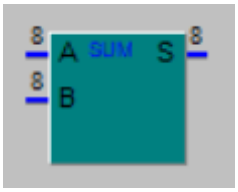
## 1 Библиотеки Standard, SignedArithm, New

Дабы улучшить читабельность кода, я добавлю некоторые сокращения, доступные в языке C:

```
#define i8  int __bits(8)
#define i16 int __bits(16)
#define i32 int __bits(32)

#define u1  unsigned __bits(1)
#define u2  unsigned __bits(2)
#define u3  unsigned __bits(3)
#define u4  unsigned __bits(4)
#define u8  unsigned __bits(8)
#define u10  unsigned __bits(10)
#define u16 unsigned __bits(16)
#define u32 unsigned __bits(32)
#define u64 unsigned __bits(64)
```

**Т. е., к примеру, int __bits(8) мы в дальнейшем будем обозначать как i8.*

**SUM:**




Устройство подает на выход сумму A и B.

C-МПА функция:
```
u8 SUM(u8 A, u8 B) {return A+B;}
```

**SIGN:**

Устройство подаёт на выход знак A(0 если число равно 0, -1 если число отрицательное и 1 если число положительное).

C-МПА функция:
```
i8 SIGN(i8 A) {return (A > 0) ? 1 : (A == 0) ? 0 : -1;}
```

**ABS:**

Устройство подаёт на выход модуль A.

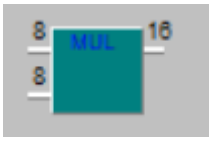
C-МПА функция:
i8 ABS(i16 A) { return (A > 0) ? A : (A*(-1));}

**FACT:**

Устройство подает на выход факториал A.

C-МПА функция:
u64 FACT(u4 A) {u64 RES=1; while (A > 1) RES *= A--; return RES;}

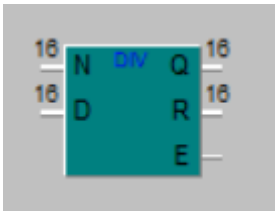
**MUL:**




Умножает значения входов A и B, и подаёт результат на выход. Размерность выхода в два раза больше чем размерность входа.

C-МПА функция:
u16 MUL(u8 A, u8 B) {return A*B;}

**DIV:**

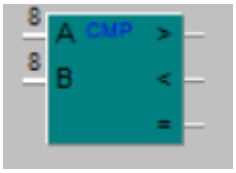



Делит значения входов N на D и подаёт частное на выход Q, а остаток на R. В случае ошибки (деление на 0) на выход E устанавливается 1. Размерность контактов произвольная. Выходы Q, R и E могут отсутствовать.

В C-МПА реализации названия контактов изменены, а именно N – A, D - B, Q – DIVRes, R – MODRes, контакт E отсутствует.

C-МПА реализация:
u16 DIVRes;
u16 MODRes;
u16 DIV(u16 A, u16 B) {return A/B;}
u16 MOD(u16 A, u16 B) {return A%B;}
void DIVResult(u16 A, u16 B)
{DIVRes = DIV(A,B); MODRes = MOD(A,B);}
(основная процедура для полноценной реализации устройства)
u32 DIVMOD(u16 A, u16 B) {return((DIV(A,B)) << 16) + MOD(A,B);}
(Функция используется если DIV имеет один выход R)

**CMP:**




Устройство производит сравнение двух значений.
G=1 если A>B
B=1 если A<B
E=1 если A=B
Разрядности A и B должны совпадать. Разрядности G,B и E всегда равны 1. Любой из контактов может отсутствовать, но необходим хотя бы один из них.
C-МПА реализация:
u1 G;
u1 E;
u1 B;
u3 CMP(u8 A, u8 B) {return (A > B) ? 4 : (A == B) ? 2 : 1;}
void CMPResult(u8 A, u8 B)
{u3 C = CMP(A,B); G = C == 4; E = C == 2; B = C == 1;}
(основная процедура для полноценной реализации устройства)

**NOT:**



Устройство подает на выход инвертированное значение входа A. Разрядность входа и выхода может быть любой, но должна совпадать.

C-МПА функция:
u1 NOT(u1 A) {return ~A;}

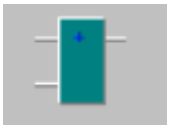
**OR2:**



Устройство реализует операцию логического "ИЛИ" над A и B.

C-МПА функция:
u1 OR2(u1 A, u1 B) {return A | B;}

**XOR2:**



Устройство реализует операцию логического исключения "ИЛИ" над A и B.

C-МПА функция:
u1 XOR2(u1 A, u 1 B) {return A ^ B;}

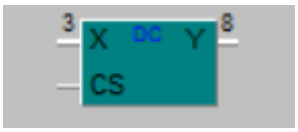
**AND2:**



Устройство реализует операцию логического "И" над A и B.

C-МПА функция:
u1 AND2(u1 A, u1 B) {return A & B;}

**DC:**




На линию Y с номером X подает 1 (все остальные линии равны 0). Разрядность Y равна 2 в степени X. При CS, установленном в 0, на всех линиях Y устанавливаются нули. CS может отсутствовать.

C-МПА функция:

u8 DC(u1 CS, u3 X) {u8 d = 1; return (CS == 0) ? 0 : d << X;}

**CD:**

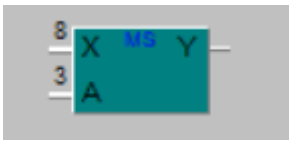



Номер самого младшего разряда X, равного 1, подается на Y. Число линий X должно быть 2 в степени числа линий Y.

C-МПА функция:

u4 CD(u8 X) {if (X == 0) return 0; u4 i = 0; while (!(X & 1)) {i++; X = X >> 1;} return (i << 1) + 1;}

**MS:**




Подает на выход Y линию X с номером A. Разрядность Y равна 1, X равна 2 в степени A.

C-МПА функция:

u1 MS(u8 X, u3 A) {return ((X >> A) & 1);}

**Следующие устройства принадлежат библиотеке Signed, поэтому могут встречаться схожие с предыдущими устройства, но вместо типа unsigned будет тип int.**

**CBW:**




Устройство расширяет переменную с 8 битов до 16.

C-МПА функция:
i16 CBW(i8 A) {return A;}

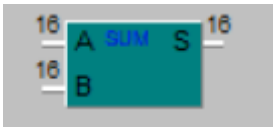
**CWD:**




Устройство расширяет  переменную с 16 битов до 32.

C-МПА функция:
i32 CWD(i16 A) {return A;}

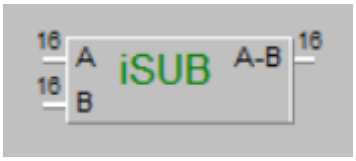
**ISUM:**




Устройство подает на выход сумму A и B.

C-МПА функция:
i16 ISUM(i16 A, i16 B) {return A+B;}

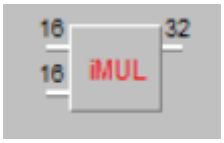
**ISUB:**




Устройство подает на выход разность A и B.

C-МПА функция:
i16 ISUB(i16 A, i16 B) {return A-B;}

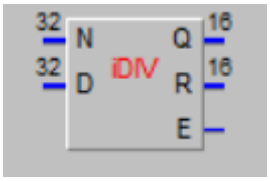
**IMUL:**




Устройство подаёт на выход произведение A и B.

C-МПА функция:
i32 IMUL(i16 A, i16 B) {return A*B;}

**IDIV:**




Устройство подаёт на выход частное N и D.

C-МПА функция:
i16 IDIV(i32 A, i32 B) {return A/B;}

**ICMP:**




Устройство производит сравнение двух значений.
IG=1 если A>B
IB=1 если A<B

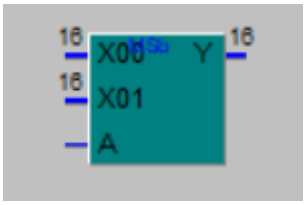
IE=1 если A=B
Разрядности A и B должны совпадать. Разрядности IG,IB и IE всегда равны 1. Любой из контактов может отсутствовать, но необходим хотя бы один из них.
C-МПА реализация:
```
u3 ICMP(i32 A, i32 B) {return (A > B) ? 4 : (A == B) ? 2 : 1;}
u1 IGT;
u1 IEQ;
u1 ILT;
void ICMPResult(i32 A, i32 B)
{u3 C = ICMP(A,B); IG = C == 4; IE = C == 2; IB = C == 1;}
```
(основная процедура для полноценной реализации устройства)

**x32Tox16:**

Устройство уменьшает размерность переменной с 32 бит до 16.

C-МПА функция:
```
u16 x32Tox16(u32 A) {return A;}
```

**MSb16x2:**




Устройство подаёт на выход значение One, если A равно 1, и Zero, если значение A равно 0.

C-МПА функция:
```
u16 MSb16x2(u16 Zero, u16 One, u1 A) {return (A) ? One : Zero;}
```

## 2 Примеры задач, решаемых с помощью данных библиотек

Условие:

На входе дано 4-разрядное число - x. Необходимо умножить это число на 5, затем к полученному результату прибавить 15.

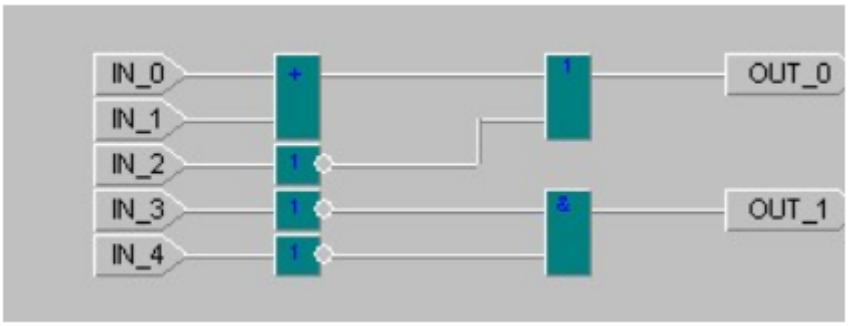
| Название | Размерность | Тип |
|---|---|---|
| x | 4 | вход |
| res | 8 | выход |

Решение С-МПА:

```
unsigned __in __bits(4) x;

unsigned __out __bits(8) res;
```

```
unsigned __bits(8) SUM(unsigned __bits(8) A, unsigned __bits(8) B) {return A+B;}
int __bits(32) IMUL(int __bits(16) A, int __bits(16) B) {return A*B;}

void main(){
unsigned a = IMUL(x,5);
a = SUM(a,15);
res=a;

}
```

Условие:
Составить программу, которая работает как приведенная ниже схема.

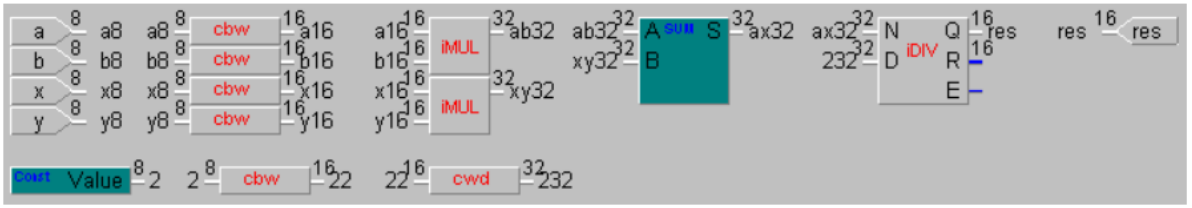



Решение С-МПА:

```
int __in __bits(1) IN_0;
int __in __bits(1) IN_1;
int __in __bits(1) IN_2;
int __in __bits(1) IN_3;
int __in __bits(1) IN_4;

unsigned __out __bits(1) OUT_0;
unsigned __out __bits(1) OUT_1;

unsigned __bits(1) XOR2 (unsigned __bits(1) A, unsigned __bits(1) B){return (A ^ B);}
unsigned __bits(1) OR2 (unsigned __bits(1) A, unsigned __bits(1) B){return (A | B);}
unsigned __bits(1) AND2 (unsigned __bits(1) A, unsigned __bits(1) B){return (A & B);}
unsigned __bits(1) NOT(unsigned __bits(1) A){return (~A);}

void main(){
```

```
unsigned xor1 = XOR2(IN_0,IN_1);
unsigned n1 = NOT(IN_2);
unsigned n2 = NOT(IN_3);
unsigned n3 = NOT(IN_4);
OUT_0 = OR2(xor1,n1);
OUT_1 = AND2(n2,n3);
}
```

Условие:
Составить программу, которая работает как приведенная ниже схема.




Значение константы равно 2.

Решение С-МПА:

```
int __in __bits(8) a;
int __in __bits(8) b;
int __in __bits(8) x;
int __in __bits(8) y;

unsigned __out __bits(16) res;

unsigned __bits(16) CBW(unsigned __bits(8) A) {return A;}
unsigned __bits(32) CWD(unsigned __bits(16) A) {return A;}
int __bits(16) ISUB(int __bits(16) A, int __bits(16) B) {return A-B;}
int __bits(32) IMUL(int __bits(16) A, int __bits(16) B) {return A*B;}
int __bits(16) IDIV(int __bits(32) A, int __bits(32) B) {return A/B;}
unsigned __bits(16) x32Tox16(unsigned __bits(32) A) {return A;}
unsigned __bits(3) ICMP(int __bits(32) A, int __bits(32) B) {return (A > B)
? 4 : (A == B) ? 2 : 1;}
unsigned __bits(16) MSb16x2(unsigned __bits(16) Zero, unsigned
__bits(16) One, unsigned __bits(1) A) {return A ? One : Zero;}
unsigned __bits(32) SUM(unsigned __bits(32) A, unsigned __bits(32) B)
{return A+B;}

void main() {
int __bits(16) a16 = CBW(a);
```

```
int __bits(16) b16 = CBW(b);
int __bits(16) x16 = CBW(x);
int __bits(16) y16 = CBW(y);
int __bits(32) ab = IMUL(a16,b16);
int __bits(32) xy = IMUL(x16,y16);
int __bits(32) s1 = SUM(ab,xy);
int __bits(32) two = 2;
int __bits(16) r = IDIV(s1,two);
res = r;
}
```

## 3 Примеры задач с некорректным созданием С-МПА программы

Условие:

На вход подаются числа a,b,c,x0,y0 (8 бит). Проверить принадлежит ли точка (x0,y0) функции y=ax^2+bx+c. Если точка принадлежит функции, то на выход res(1 бит) подать 1, в противном случае, на выход res подать 0.

| Название | Размерность | Тип |
|---|---|---|
| a | 8 | вход |
| b | 8 | вход |
| c | 8 | вход |
| x0 | 8 | вход |
| y0 | 8 | вход |
| res | 1 | выход |

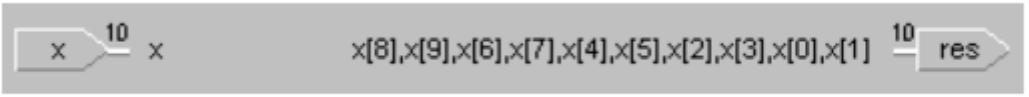
Некорректное решение С-МПА:

```
int __in __bits(8) a;
```

```
int __in __bits(8) b;
int __in __bits(8) c;
int __in __bits(8) x0;
int __in __bits(8) y0;

unsigned __out __bits(1) res;

unsigned __bits(16) CBW(unsigned __bits(8) A) {return A;}
unsigned __bits(32) CWD(unsigned __bits(16) A) {return A;}
int __bits(16) ISUB(int __bits(16) A, int __bits(16) B) {return A-B;}
int __bits(32) IMUL(int __bits(16) A, int __bits(16) B) {return A*B;}
int __bits(16) IDIV(int __bits(32) A, int __bits(32) B) {return A/B;}
unsigned __bits(16) x32Tox16(unsigned __bits(32) A) {return A;}
unsigned __bits(3) ICMP(int __bits(32) A, int __bits(32) B) {return (A > B)
? 4 : (A == B) ? 2 : 1;}
unsigned __bits(16) MSb16x2(unsigned __bits(16) Zero, unsigned
__bits(16) One, unsigned __bits(1) A) {return A ? One : Zero;}
unsigned __bits(16) SUM(unsigned __bits(16) A, unsigned __bits(16) B)
{return A+B;}

void main() {
int a16 = CBW(a);
int b16 = CBW(b);
int c16 = CBW(c);
int x16 = CBW(x0);
int y16 = CBW(y0);
int xx = IMUL(x16,x16);
int axx = IMUL(a16,xx);
int bx = IMUL(b16,x16);
int s1 = SUM(axx,bx);
int s2 = SUM(s1,c16);


int n = 0;
if((ICMP(s2,y16)==2)){n = 1;}
res=MSb16x2(0,1,n);

}
```

Описание ошибки: **Нельзя использовать оператор IF - его нет в HLCCAD.**

Условие:
Составить программу, которая работает как приведенная ниже схема.

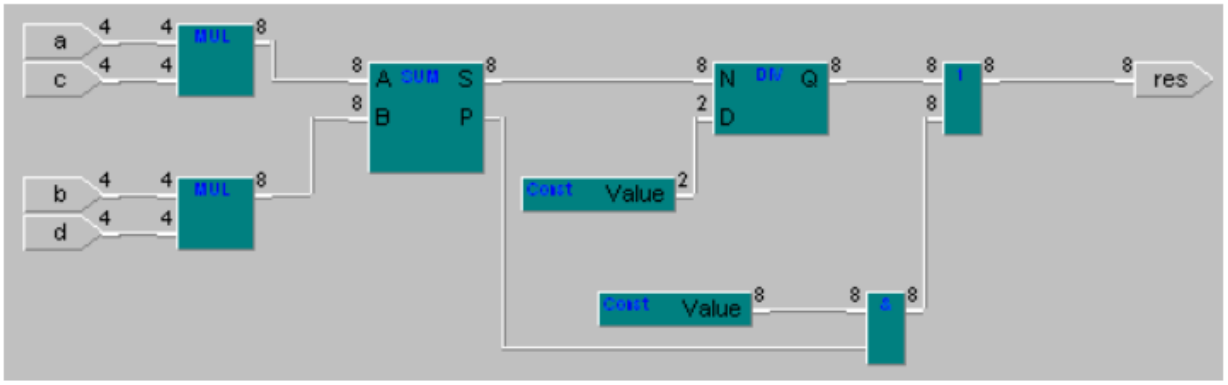



Некорректное решение С-МПА:

```
unsigned __in __bits(10) x;

unsigned __out __bits(10) res;

unsigned __bits(1) BIT(unsigned __bits(10) x,unsigned __bits(4) n){ return x>>n;}

void main()

{
 unsigned a=BIT(x,1);
  a+=BIT(x,0)*2;
  a+=BIT(x,3)*2*2;
  a+=BIT(x,2)*2*2*2;
  a+=BIT(x,5)*2*2*2*2;
  a+=BIT(x,4)*2*2*2*2*2;
  a+=BIT(x,7)*2*2*2*2*2*2;
  a+=BIT(x,6)*2*2*2*2*2*2*2;
  a+=BIT(x,9)*2*2*2*2*2*2*2*2;
  a+=BIT(x,8)*2*2*2*2*2*2*2*2*2;
  res = a;
 }
```

Описание ошибки: **Для HLCCAD нельзя писать *2, надо использовать умножитель MUL, нельзя писать a+=, надо использовать сумматор.**

Условие:
Составить программу, которая работает как приведенная ниже схема.




Значения констант сверху вниз: 2, 128.

Некорректное решение С-МПА:

```
unsigned int __in __bits(4) a;
unsigned int __in __bits(4) b;
unsigned int __in __bits(4) c;
unsigned int __in __bits(4) d;

unsigned int __out __bits(8) res;

unsigned int __bits(16) SUM(unsigned int __bits(8) A, unsigned int __bits(8) B) {return A+B;}
unsigned __bits(8) DIV(unsigned __bits(16) A, unsigned __bits(2) B) {return A/B;}
unsigned __bits(8) MUL(unsigned __bits(4) A, unsigned __bits(4) B) {return A * B;}
unsigned __bits(8) AND2(unsigned __bits(8) A, unsigned __bits(8) B) {return A & B;}
unsigned __bits(8) OR2(unsigned __bits(8) A, unsigned __bits(8) B) {return A | B;}

void main() {

unsigned int __out __bits(8) mul1 = MUL(a,c);
unsigned int __out __bits(8) mul2 = MUL(b,d);
unsigned int __out __bits(16) sum = SUM(mul1,mul2);
unsigned int __out __bits(8) p = ((b > 0) && (a > -b)) || ((b < 0) && (a < -b)) ? 1 : 0;
unsigned int __out __bits(2) const2 = 2;
unsigned int __out __bits(16) const128 = 128;
unsigned int __out __bits(8) div = DIV(sum,const2);
unsigned int __out __bits(8) and = AND2(const128,p);
```

```
res = OR2(div,and);
}
```

Описание ошибки: **Надо использовать только блоки HLCCAD.**

**Этого там нет:** unsigned int __out __bits(8) p = ((b > 0) && (a > -b)) || ((b < 0) && (a < -b)) ? 1 : 0;

## 4 Примеры задач, не решаемых только с помощью библиотек Standard, SignedArithm, New

Условие: Составить программу, которая работает как приведенная ниже схема.



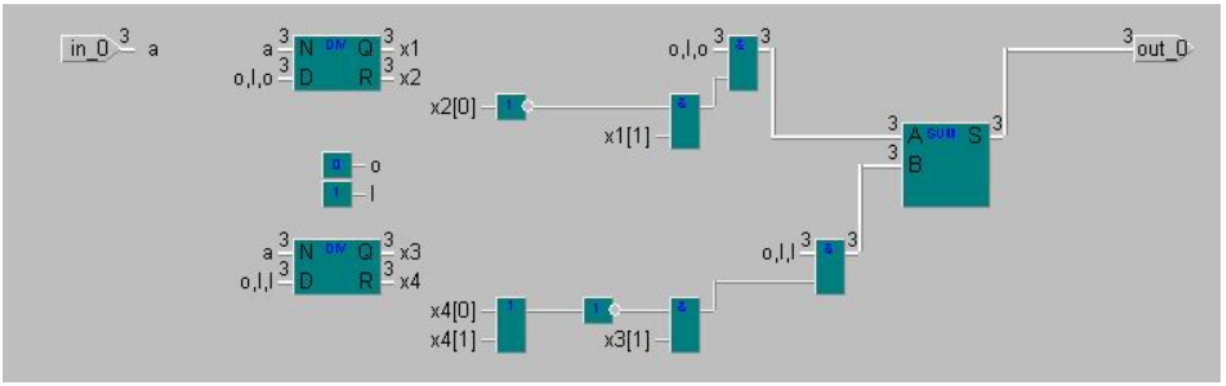

Решение С-МПА:

```
int __in __bits(32) In;
```

```
int __out __bits(1) Out;

unsigned __bits(1) BIT(unsigned __bits(32) x,unsigned __bits(10) n){ return x>>n;}

unsigned __bits(1) XOR1(unsigned __bits(32) A) {unsigned z; unsigned x; x=BIT(A,0); for(int i=1;i<32;i++){z=BIT(A,i); x=x^z;}
return x;}

void main(){
  Out = XOR1(In);
}
```

Функции, отсутствующие в библиотеках Standard, SignedArithm, New**:**
BIT, XOR1.

Условие:
Составить программу, которая работает как приведенная ниже схема.



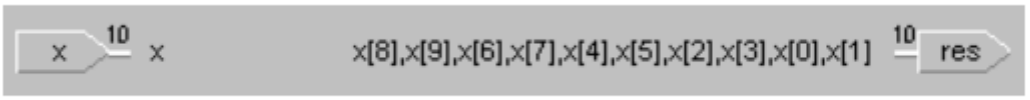

Решение С-МПА:

```
unsigned __in __bits(3) in_0;

unsigned __out __bits(3) OUT_0;

unsigned __bits(1) BIT(unsigned __bits(3) x,unsigned __bits(1) n){ return x>>n;}

unsigned __bits(3) ANDp8(unsigned __bits(3) x, unsigned __bits(1) y)
{unsigned a; unsigned b; unsigned c; a=x&y; b=(x>>1)&y; c=(x>>2)&y;
      return a+b*2+c*4;}
```

```
unsigned __bits(3) DIVRes;
unsigned __bits(3) MODRes;
unsigned __bits(3) DIV(unsigned __bits(3) A, unsigned __bits(3) B) {return A/B;}
unsigned __bits(3) MOD(unsigned __bits(3) A, unsigned __bits(3) B) {return A%B;}
unsigned __bits(3) SUM(unsigned __bits(3) A, unsigned __bits(3) B) {return A+B;}
unsigned __bits(1) NOT(unsigned __bits(1) A) {return ~A;}
unsigned __bits(1) AND2(unsigned __bits(1) A, unsigned __bits(1) B) {return A & B;}
unsigned __bits(1) OR2(unsigned __bits(1) A, unsigned __bits(1) B) {return A | B;}
void DIVResult(unsigned __bits(3) A, unsigned __bits(3) B) {DIVRes = DIV(A,B); MODRes = MOD(A,B);}


void main()
{
  DIVResult(in_0, 2);
  unsigned a = NOT(BIT(MODRes,0));
  a = AND2(a,BIT(DIVRes,1));
  a = ANDp8(2,a);
  DIVResult(in_0, 3);

        unsigned r = BIT(MODRes,0);
        unsigned e = BIT(MODRes,1) ;

  unsigned q = OR2(r,e);
  unsigned   b =     NOT(q);
  b = AND2(b,BIT(DIVRes,1));
  b = ANDp8(3,b);
  OUT_0 = SUM(a,b);
 }
```

Функции, отсутствующие в библиотеках Standard, SignedArithm, New**:** ANDp8, BIT.

Условие:
Составить программу, которая работает как приведенная ниже схема.




Решение С-МПА:

```
unsigned __in __bits(10) x;

unsigned __out __bits(10) res;

unsigned __bits(1) BIT(unsigned __bits(10) x,unsigned __bits(4) n){ return x>>n;}

unsigned __bits(10) BUS(unsigned __bits(1) x1,unsigned __bits(1) x2,unsigned __bits(1) x3,unsigned __bits(1) x4,unsigned __bits(1) x5,unsigned __bits(1) x6,unsigned __bits(1) x7,unsigned __bits(1) x8,unsigned __bits(1) x9,unsigned __bits(1) x10)
{unsigned __bits(10) a; a=x1; a=a<<1; a+=x2; a=a<<1; a+=x3; a=a<<1; a+=x4; a=a<<1; a+=x5; a=a<<1; a+=x6; a=a<<1; a+=x7; a=a<<1; a+=x8; a=a<<1; a+=x9; a=a<<1; a+=x10; return a;}

void main()
{
 unsigned x1 = BIT(x,8);
 unsigned x2 = BIT(x,9);
 unsigned x3 = BIT(x,6);
 unsigned x4 = BIT(x,7);
 unsigned x5 = BIT(x,4);
 unsigned x6 = BIT(x,5);
 unsigned x7 = BIT(x,2);
 unsigned x8 = BIT(x,3);
 unsigned x9 = BIT(x,0);
 unsigned x10 =BIT(x,1);

 res = BUS(x1,x2,x3,x4,x5,x6,x7,x8,x9,x10);
 }
```

Функции, отсутствующие в библиотеках Standard, SignedArithm, New**:** BUS, BIT.

## 5 Библиотека EXTENDED

**DC:**




На линию Y с номером X подает 1 (все остальные линии равны 0). Разрядность Y равна 2 в степени X. При CS, установленном в 0, на всех линиях Y устанавливаются нули. CS может отсутствовать.

C-МПА функция:

u8 DC(u3 X) {u8 d = 1; return  d << X;}

**OR3:**



Устройство реализует операцию логического "ИЛИ" над A, B и C.

C-МПА функция:
u1 OR3(u1 A, u1 B, u1 C) {return A | B | C;}

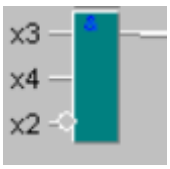
**OR4:**



Устройство реализует операцию логического "ИЛИ" над A, B, C и D.

C-МПА функция:
u1 OR4(u1 A, u1 B, u1 C, u1 D) {return A | B | C | D;}

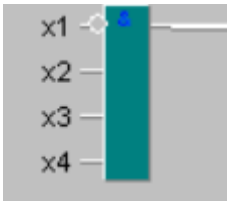
**AND3:**




Устройство реализует операцию логического "И" над A, B и C.

C-МПА функция:
u1 AND3(u1 A, u1 B, u1 C) {return A & B & C;}

**AND4:**




Устройство реализует операцию логического "И" над A, B, C и D.

C-МПА функция:
u1 AND4(u1 A, u1 B, u1 C, u1 D) {return A & B & C & D;}

**BLOCK:**

**Корпус:**

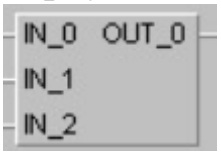
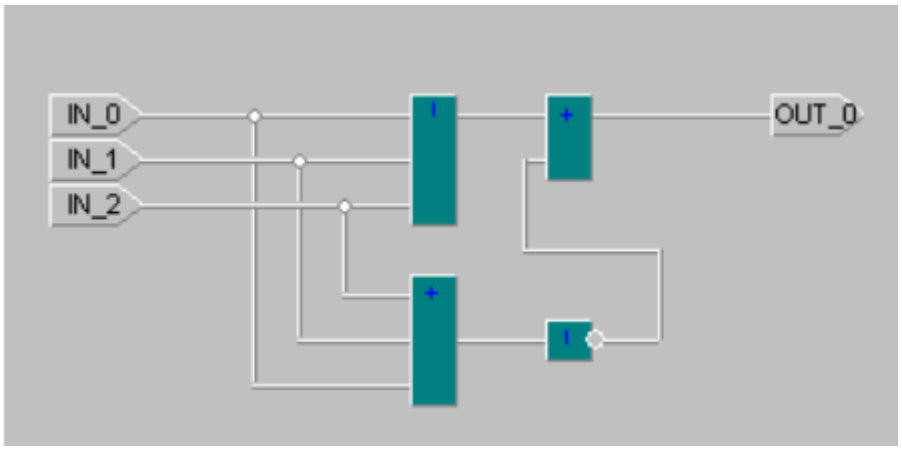



**Схема:**

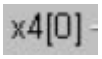



Устройство выполняет операции над входными контактами IN_0, IN_1, IN_2 по схеме выше.

C-МПА функция:
u1 BLOCK(u1 IN_0, u1 IN_1, u1 IN_2)

{u1 b=NOT(XOR3(IN_0,IN_1,IN_2)); u1 a=OR3(IN_0,IN_1,IN_2);
return XOR2(a,b);}

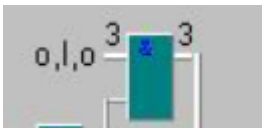
**BIT:**




Устройство подаёт на выход n-ный бит числа x.

C-МПА функция:
u1 BIT(u3 x, u3 n){return x>>n;}

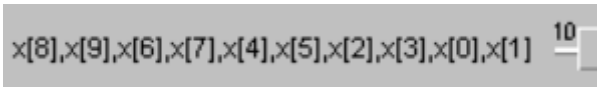
**ANDp8:**




Устройство работает следующим образом: "Если один из входов имеет разрядность больше 1, а другой 1, то выполняется логическое И всех линий первого входа с линией второго." Она работает только с входным параметром размером в 3 бита.

C-МПА функция:
u3 ANDp8(u3 x, u1 y)
{u1 a; u2 b; u3 c; a=x&y; b=(x>>1)&y; c=(x>>2)&y; return a+b*2+c*4;}

**BUSx10:**

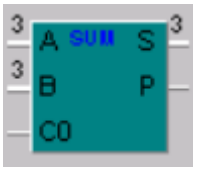



Устройство объединяет биты в десятеричное число.

C-МПА функция:

u10 BUSx10(u1 x1, u1 x2, u1 x3, u1 x4, u1 x5, u1 x6, u1 x7, u1 x8, u1 x9, u1 x10){u10 a; a=x1; a=a<<1; a+=x2; a=a<<1; a+=x3; a=a<<1; a+=x4; a=a<<1; a+=x5; a=a<<1; a+=x6; a=a<<1; a+=x7; a=a<<1; a+=x8; a=a<<1; a+=x9; a=a<<1; a+=x10; return a;}

**Полный по HLCCAD SUM:**




Устройство подает на выход S сумму A,B и C0. В случае переполнения устанавливает P в 1 (иначе P - 0). Разрядность A, B и S любая, но должна совпадать. Разрядность C0 и P равна 1. Могут отсутствовать C0,S и P.

В дальнейшей реализации C0 заменено на z, A заменено на x, B заменено на y, P на bt(bit), a S на resx3.

C-МПА реализация:

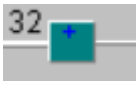
u4 SUM(u3 A, u3 B) {return A+B;}

u1 bt;

u3 resx3;

void SUMResultx3(u3 x, u3 y, u1 z) {u4 i; i=SUM(x,y); resx3=SUM(i,z); u4 t; t=SUM(x,y); bt=t>>3;}

**XOR1:**




Устройство подаёт на выход поразрядное И над входными линиями.

Также для корректной работы устройства потребовалось использовать функцию BIT.

C-МПА функция:

u1 BIT(u32 x, u10 n){ return x>>n;}

u1 XOR1(u32 A) {u1 z; u1 x; x=BIT(A,0); for(int i=1;i<32;i++){z=BIT(A,i); x=x^z;} return x;}

# 6 Дальнейшие направления работы

## 6.1 Пошаговая генерация HLCCAD-схемы по C-МПА программе, выпол-ненной на основе библиотек

В дальнейшем планируется разработка некоторого визуального интерфейса в самой HLCCAD для оперирования над IA. Этот интерфейс, разработанный, к примеру, на Delphy 7, поможет генерировать пользователю схему HLCCAD для решения некоторой задачи поэтапно, что облегчит студенту обучение. Также можно будет добавить и другие функции как к примеру генерация схемы по условию.

## 6.2 Обучение программ AI генерации C-МПА программ по условию задачи

На данном этапе моими сокурсниками была реализована генерация решения задачи на C++ по ее условию, однако нам предстоит научится преобразовать эти решения к виду уже реализованных функций C-МПА для дальнейшего преобразования в HLCCAB-схемы. Можно также попробовать поработать с AI, что бы вместо промежуточного решения на C++ было сразу

сделано решение при помощи функций библиотек Standard, SignedArithm, New, EXTENDED.

### 6.3 Развитие библиотек

Хоть нами и было создано не мало функций и библиотек, однако в дальнейшем наверняка появится потребность в создании новых. Они могут быть реализованы как на базе уже имеющихся, так и созданы уникально. И среди этих функций встречается много повторяющихся как, к примеру, OR. Функция OR на данном этапе существует в нескольких экземплярах: для разного количества входных контактов, т.е. OR2, OR3, OR4. Это сильно перегружает библиотеку, а также крайне неудобно. В языке C++ для решения такой проблемы существует переполнение функций, когда для функции с одним и тем же названием используется разное количество входных переменных, и в зависимости от этого вызывается нужная функция. В C-МПА на данном этапе с этим проблемы и переполнение функций не работает. Решение этой проблемы является частью развития нашего проекта.

Развитие библиотек C-МПА в конечном итоге должно прийти к объединению в полноценные библиотеки C, подключаемые при помощи include. Это значительно упростит работу по обучению AI.

## ЗАКЛЮЧЕНИЕ

В заключение данной курсовой работы мы видим, что искусственный интеллект становится ключевым фактором в различных сферах человеческой деятельности. Проникновение технологий искусственного интеллекта в медицину, образование, промышленность и научные исследования открывает новые возможности и перспективы для общества. В частности, в ходе данной работы был проделан значительный шаг в развитии искусственного интеллекта путем создания функций на C-МПА, аналогичных устройствам HLCCAD.

Данная курсовая работа предоставляет обзор корректных и некорректных решений задач, а также подробные объяснения принципов работы всех функций HLCCAD. Это позволяет углубиться в суть проблемы и обеспечивает основательное понимание особенностей функционирования систем искусственного интеллекта. Работа с IA представляет собой сложную задачу, требующую постоянного совершенствования и улучшения алгоритмов и методов.

Таким образом, развитие искусственного интеллекта в современном мире представляет собой важный этап технологической революции, и данная

работа вносит свой вклад в это направление, предоставляя новые инструменты и практические решения для улучшения эффективности работы с искусственным интеллектом.

На данный момент были достигнуты следующие цели:

-Созданы библиотеки Extended, SignedArithm, New

-IA способен понимать условие задачи и создавать решение на языке C++

-IA способен воспринимать C-МПА код, симулирующий работу HLCCAD, и при помощи него создавать на корпусе устройства, соединяя его при помощи невидимых шин

Предстоящие для достижения цели:

-Объединение имеющихся функций а отдельные полноценные библиотеки, подключаемые при помощи include

-Разработка пользовательского интерфейса для взаимодействия со всеми функциями IA

-Реализация переполнения функций, аналогичному в языке C++

-Полноценная генерация схемы HLCCAD по условию задачи

ПРИЛОЖЕНИЕ А

ПРИЛОЖЕНИЕ В

**Результат проверки на плагиат**

2024-01-08 03:47:25

**Уникальность текста: 87%**

| Источник | Сходство % |
|---|---|
| https://www.chegg.com/homework-help/questions-and-answers/int-subf-int-int-b-return-b-cubef-int-return-int-minf-int-in… | 3% |
| https://otvet.mail.ru/question/223808720 | 3% |
| https://konkurs-klass.ru/rasskazy_2023 | 3% |
| https://www.sao.ru/hq/sts/linux/book/bogatyrev_c_unix/gl_1_4.shtml | 3% |
| https://stackoverflow.com/a/24768663 | 2% |
| https://stackoverflow.com/questions/39202616/difference-between-sort-sortfunctiona-breturn-a-b-and-sortfunctiona | 2% |
| https://ptpit.ru/learn/JSADV/0101.htm | 2% |
| https://forum.freecodecamp.org/t/arr-sort-a-b-a-b-explanation/167677 | 2% |
| https://forum.freecodecamp.org/t/need-an-explanation-why-we-are-returning-a-b/187258 | 2% |
| https://forum.freecodecamp.org/t/sort-an-array-alphabetically-using-the-sort-method-what-does-the-return-statements-i… | 2% |
| https://forum.freecodecamp.org/t/trouble-with-sort-compare-func-with-numbers-why-doesnt-it-return-a-b/388417 | 2% |
| https://umedp.ru/articles/fibrillyatsiya_predserdiy_slozhnye_sluchai_iz_praktiki_kardiologa.html | 2% |
| https://pythonist.ru/return-in-python/ | 2% |
| https://www.reddit.com/r/learnjavascript/comments/gjibw9/how_does_the_sortfunctionab_return_ab_function/ | 2% |
| https://blog.hubspot.com/website/sorting-javascript | 2% |