

**ВОЛГОГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ**

**Факультет электроники и вычислительной техники
Кафедра ЭВМ и систем**

**Семестровая работа
по дисциплине Архитектура ЭВМ
«Построение простого магистрального процессора»
Вариант - 47 (101111)**

Выполнил

Группа

Проверил

Титов А.К.

ИВТ-261

к.т.н., Андреев
А.Е.

Вариант № 47

Задание:

Пункт	1 вариант	2 вариант	3 вариант	4 вариант
1. Используемые варианты адресации (у всех – регистровая и прямая) + ...	Косвенная регистровая, непосредственная	Косвенная регистровая, косвенная регистровая по базе (индексная)	Непосредственная, косвенная регистровая по базе (индексная) с непосредственным смещением	Косвенная регистровая по базе (индексная) с масштабируемой базой и непосредственным смещением
2. Адресность команд	Двухадресные (типа add r1, r2)	Трехадресные (логич арифм (типа add r1, r2, r3))		
3. Операции АЛУ у всех : ADD + ...	+ SHL (влево логич.), NAND, INC	+ SHRA (вправо арифм.), NAND, SET	+ SHLA (влево арифм.), NOR, DEC	+ SHR (вправо логич.), NOR, EQU
4. Тип сдвигателя в АЛУ	Комбинационный на любое число разрядов в одну сторону	Монтажный сдвиг на 1 разряд в обе стороны	Монтажный сдвиг на 1 разряд в одну сторону	Комбинационный на любое число разрядов в обе стороны
5. Адресная арифметика	На общем АЛУ	На отдельном АЛУ		
6. Состав команд – у всех (процессора) : LD, ST, ADD, SHRA + ...	JZ, NAND, JMP, SHL	JC, NOR, INC, SLC	JN, NAND, SUB, DEC	JE, NOR, JMP, XCH
7. Логика обработки разных форматов одних и тех же команд	Микро-программная	Комбинационная		

Псевдокод микропрограммы

START

- 1) Прочитать из памяти по адресу PC в регистр IR
- 2) DECODE

LD:

- 1) Увеличить значение PC на 1
- 2) Загрузить по адресу PC из памяти второй бит команды в регистр AIR
- 3) Копировать данные из регистра, номер которого Ry в регистр INDEX
- 4) Загрузить из памяти данные в регистр R1 команды
- 5) JMP END

ST:

- 1) Увеличить значение PC на 1
- 2) Загрузить по адресу PC из памяти второй бит команды в регистр AIR
- 3) Копировать данные из регистра, номер которого Ry в регистр INDEX
- 4) Записать в память данные из регистра R1 команды
- 5) JMP END

ADD:

- 1) Увеличить значение PC на 1
- 2) Загрузить по адресу PC из памяти второй бит команды в регистр AIR
- 3) Выполнить сложение
- 4) JMP END

NOR:

- 1) Увеличить значение PC на 1
- 2) Загрузить по адресу PC из памяти второй бит команды в регистр AIR
- 3) Выполнить NOR
- 4) JMP END

SHRA:

- 1) Выполнить SHRA
- 2) JMP END

JE:

- 1) Увеличить значение PC на 1
- 2) Загрузить по адресу PC из памяти второй бит команды в регистр AIR
- 3) Записать в PC содержимое AIR
- 4) JMP START

JMP:

- 1) Увеличить значение PC на 1
- 2) Загрузить по адресу PC из памяти второй бит команды в регистр AIR
- 3) Записать в PC содержимое AIR
- 4) JMP START

XCHG:

- 1) Увеличить значение PC на 1
- 2) Загрузить по адресу PC из памяти второй бит команды в регистр AIR
- 3) Записать значение первого регистра в temp
- 4) Записать значение второго регистра в первый регистр
- 5) Записать значение temp во второй регистр
- 6) JMP END

END:

- 1) INC(PC)
- 2) JMP START

Форматы команд

Наименование команды	Формат команды	Первый байт команды						Второй байт команды					
LD, ST	1	КОП		ФК		Rx		address					
	2	КОП		ФК		Rx		Ry					
	3	КОП		ФК		Rx		Ry	Rz				
JE, JMP		КОП						address					
XCHG		КОП				Rx		Ry					
ADD, NOR		КОП				Rx		Ry	Rz				
SHRA		КОП				Rx							

ФК — формат команды
КОП — код операции
Rx, Ry, Rz – регистры ($R_x = R_y + R_z$)

КОД операции	Аббревиатура операции
000	LD
001	ST
010	ADD
011	NOR
100	SHRA
101	XCHG
110	JE
111	JMP

Формат команды	Значение поля ФК
Rx, Ry	00
Rx, Ry, Rz	01
Rx, address	10

Формат микрокоманды

A T S	ADDR _ TYPE	A M X	NA			B M X	NB			C M X	NC			КОП АЛУ			W F	M W	M R	ACode		JMP a d d r e s s					

Описание микрокоманды

Наименование группы битов	Число битов	Пояснения
ATS (addr_type source)	1	Выбор источника для выбора вида адресации (команда /микрокоманда)
ADDR_TYPE	2	Тип адресации
AMX	1	Выбор источника для регистра А (команда/микрокоманда)
NA	3	Номер регистра А
BMX	1	Выбор источника для регистра В (команда/микрокоманда)
NB	3	Номер регистра В
CMX	1	Выбор источника для регистра С (команда/микрокоманда)
NC	3	Номер регистра С
КОП АЛУ	3	Код операции АЛУ
WF	1	Запись флага EF
MW	1	Запись в память
MR	1	Чтение из памяти
ACode	2	Способ выбора следующей микрокоманды
JMP address	6	Адрес перехода (для JMP и JE)
Всего	29	

Значения битов микрокоманды

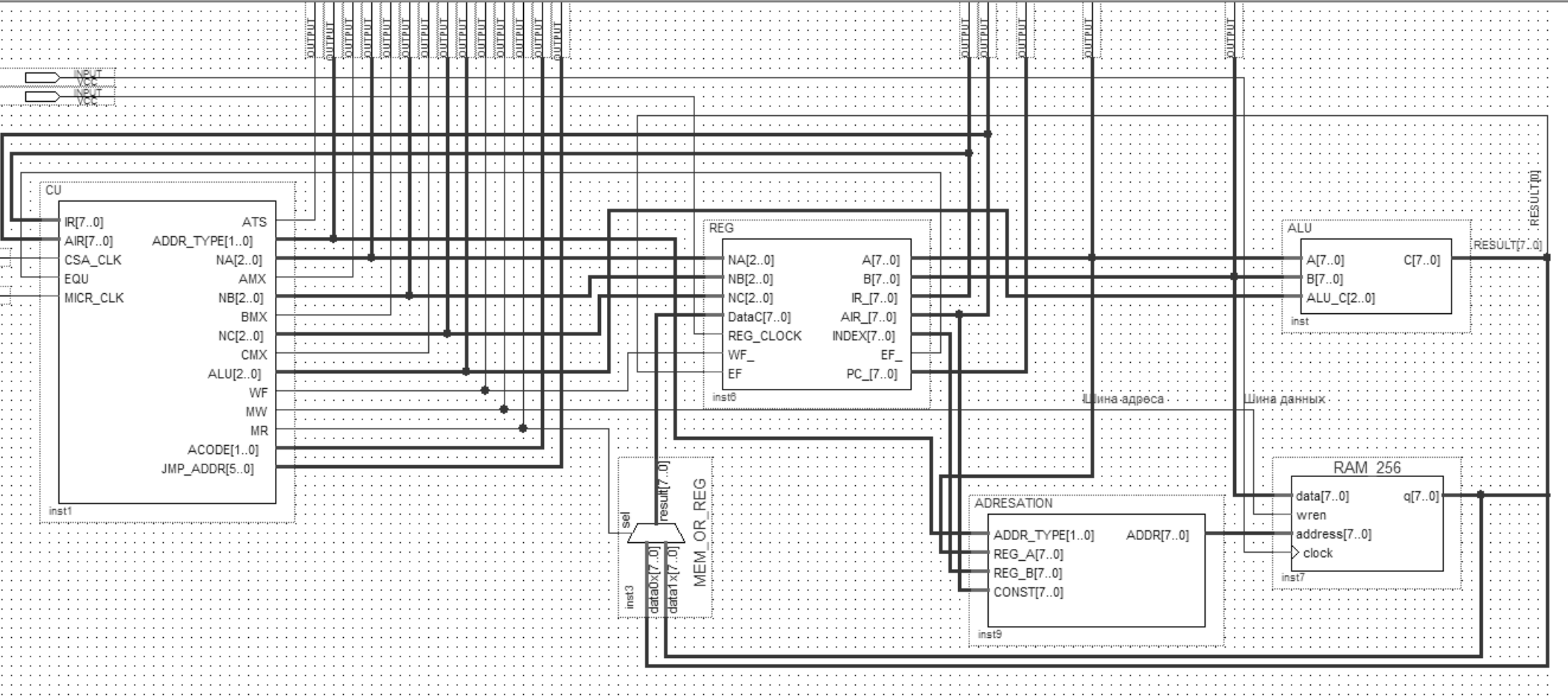
Наименование группы битов	Значение	Пояснение
ATS	0	Источник типа адресации - команда
	1	Источник типа адресации - микрокоманда
ADDR TYPE	00	Регистровая адресация (LD Rx, Ry)
	01	Индексная адресация (LD Rx, Ry , Rz)
	11	Прямая адресация (LD Rx, address)
AMX, BMX, CMX	0	Номер регистра А берется из команды
	1	Номер регистра А берется из микрокоманды
NA, NB, NC	000	REG 0
	001	REG 1
	010	REG 2
	011	REG 3
	100	PC (счетчик программы)
	101	IR (первый байт команды)
	110	AIR (второй байт команды)
	111	INDEX (для индексной адресации)
КОП АЛУ	000	ADD
	001	EQU
	010	NOR
	011	SHR
	100	INC(1) (увеличение на 1)
	101	INC(2) (увеличение на 2)
	110	Получить значение регистра А без изменений (для поддержки MOV)
	111	Получить значение регистра В без изменений (для поддержки MOV)
WF	0	Не обновлять регистр флага EF при этой операции
	1	Обновить регистр флага EF при этой операции
MW	0	Память закрыта на запись
	1	Память открыта на запись
MR	0	Память закрыта на чтение
	1	Память открыта на чтение
ACode	00	NEXT
	01	DECODE
	10	JE
	11	JMP
JMP address	000000	Адрес перехода (для JMP и JE)

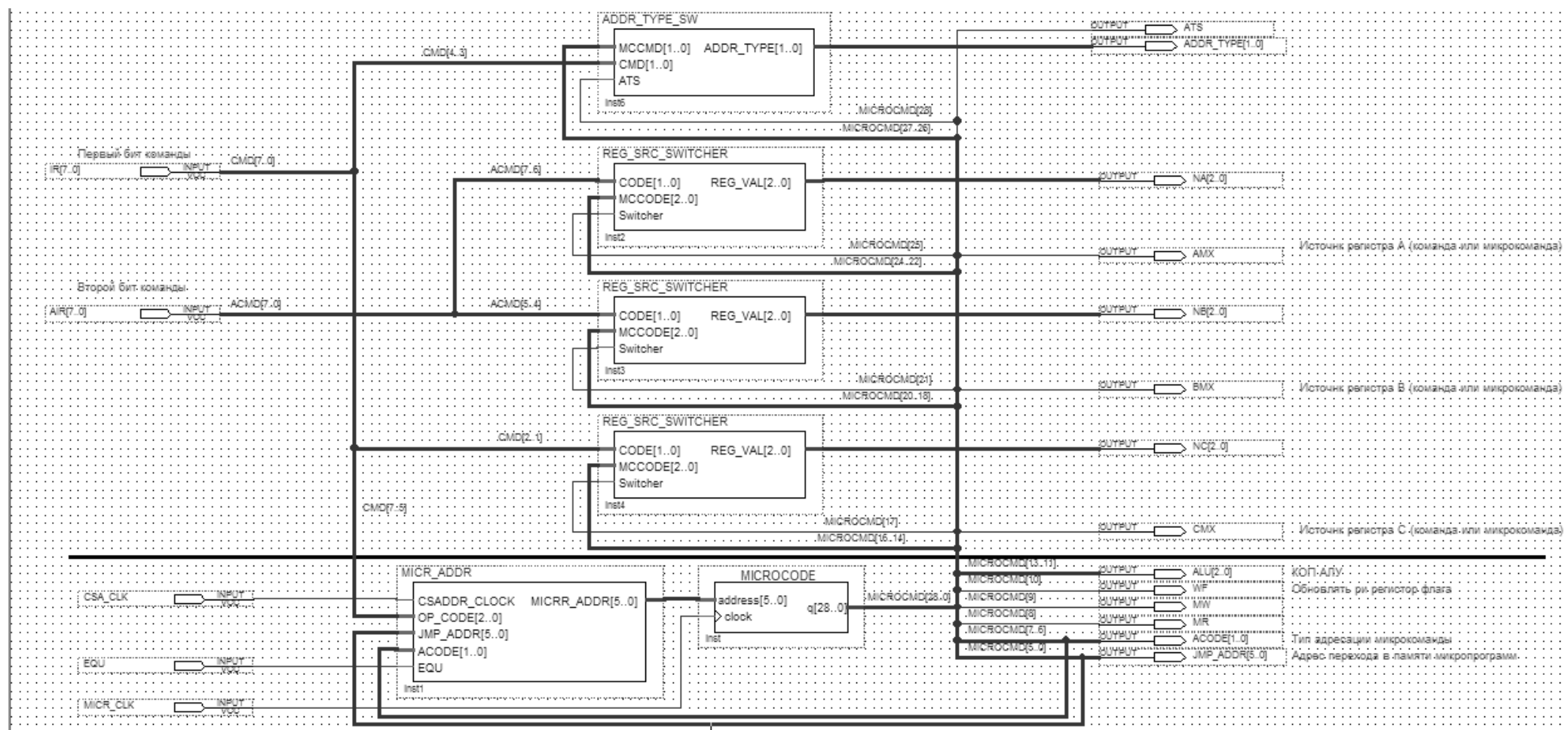
Микрокод

№	A T S	ADDR TYPE		A M X	NA			B M X	NB			C M X	NC			КОП АЛГ			W F	M W	M R	ACode		JMP address							
	1	0	0	1	1	0	0	1	1	0	0	1	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0		
1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0			
2	1	0	0	1	1	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0			
3	1	0	0	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0			
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1			
5	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0			
6	1	0	0	1	1	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0			
7	1	0	0	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0			
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1			
9	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0			
10	1	0	0	1	1	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0			
11	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1			
12	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0			
13	1	0	0	1	1	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0			
14	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1	1	1	1			
15	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	1	0	0	0	1	1	1	1	1	1	1			
16	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0			
17	1	0	0	1	1	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0			
18	1	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0			
19	1	0	0					0	0	0	0	1							0	0	0	0	0	0	0	0	0	0			
20	1	0	0	1	1	1	1									1	1	0	0	0	0	1	1	1	1	1	1	1			
21	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0			
22	1	0	0	1	1	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0			

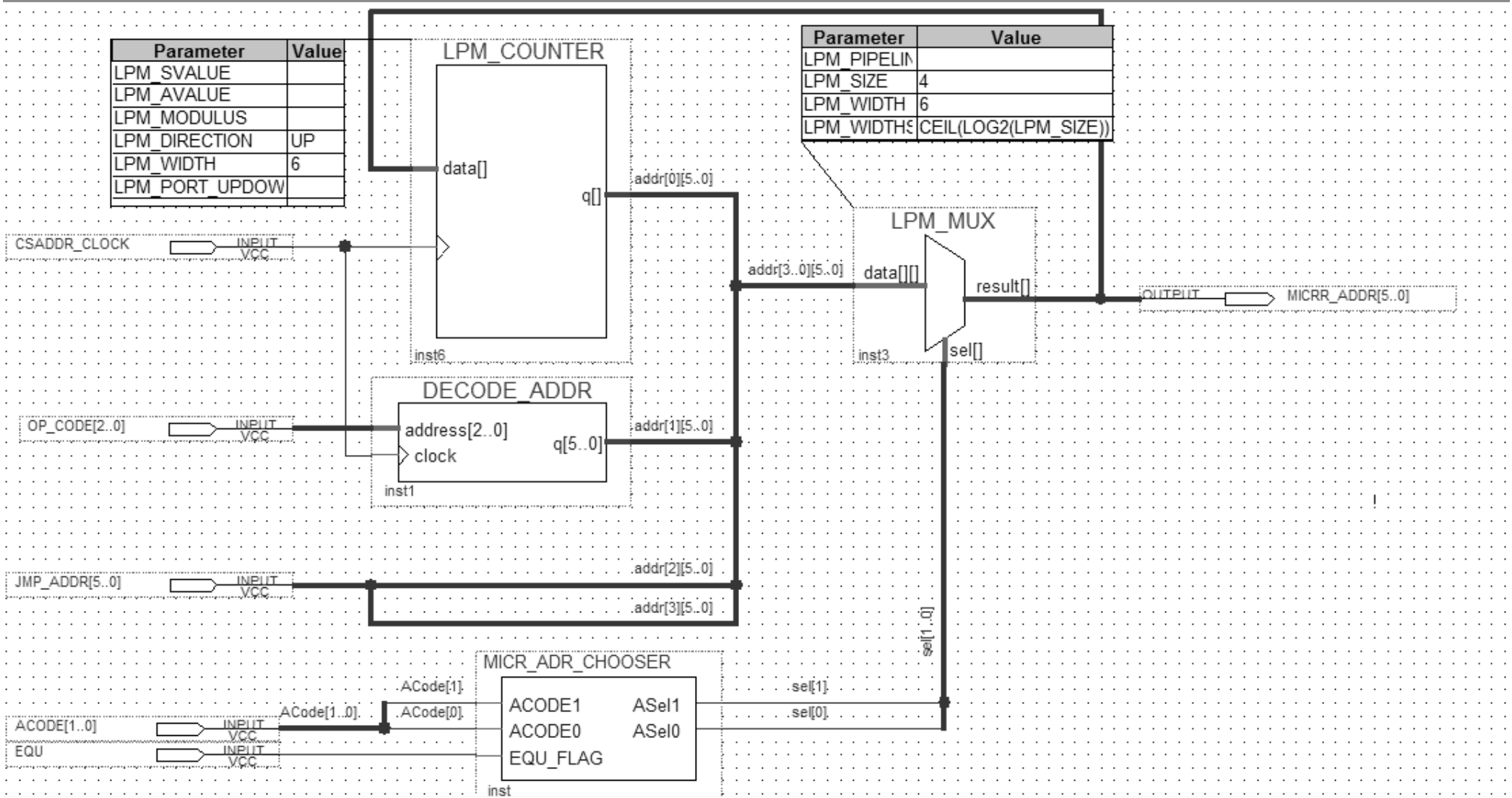
№	A T S	ADDR TYPE		A M X	NA			B M X	NB			C M X	NC			КОП АЛЮ			W F	M W	M R	ACode		JMP address							
23	1	0	0	1	1	0	0	1	0	0	0	1	1	1	0	1	1	0	0	0	0	1	1	1	1	1	1	1	1		
24	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0		
25	1	0	0	1	1	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0		
26	1	0	0	1	1	0	0	1	0	0	0	1	1	1	0	1	1	0	0	0	0	1	1	1	1	1	1	1	1		
27																															
28																															
...																															
63																															
64	1	0	0	1	1	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0		

CPU

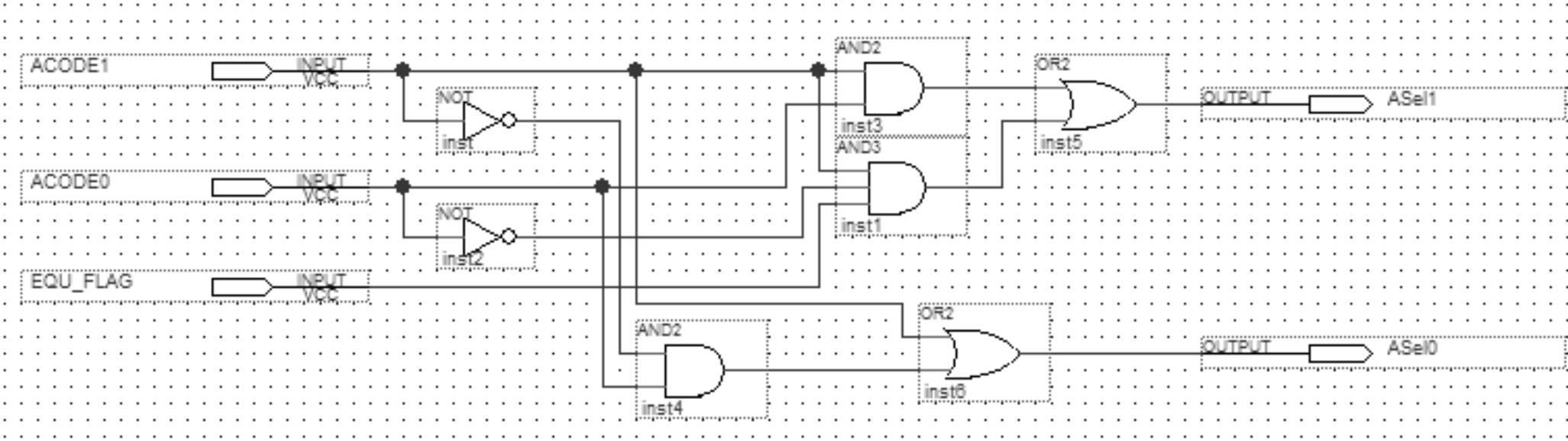




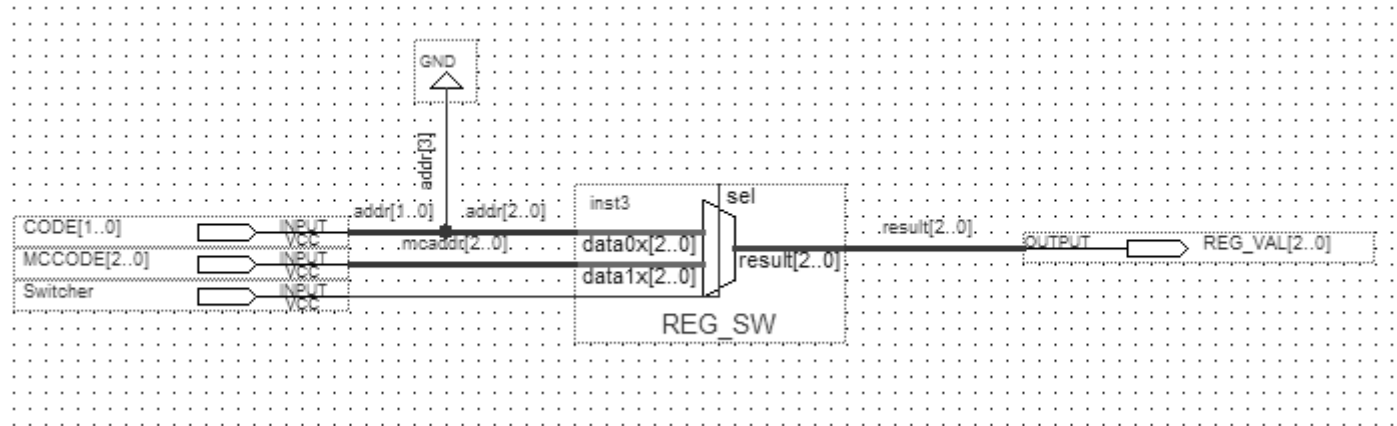
MICR_ADDR



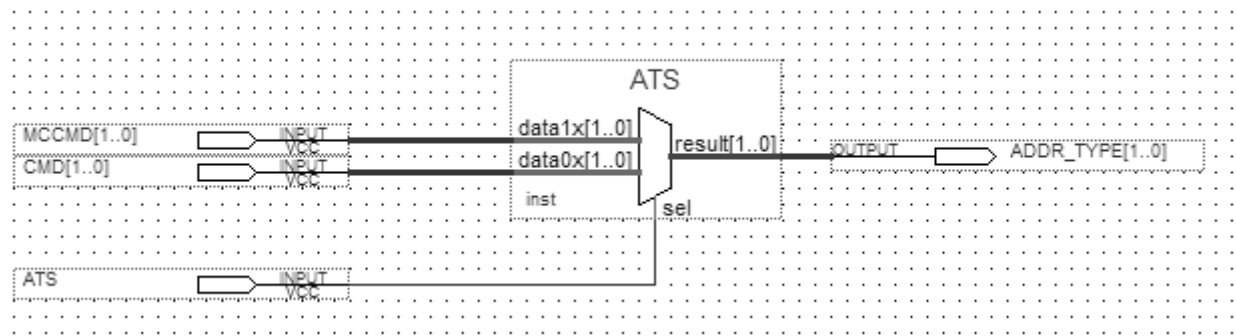
MICR_ADDR_CHOOSER



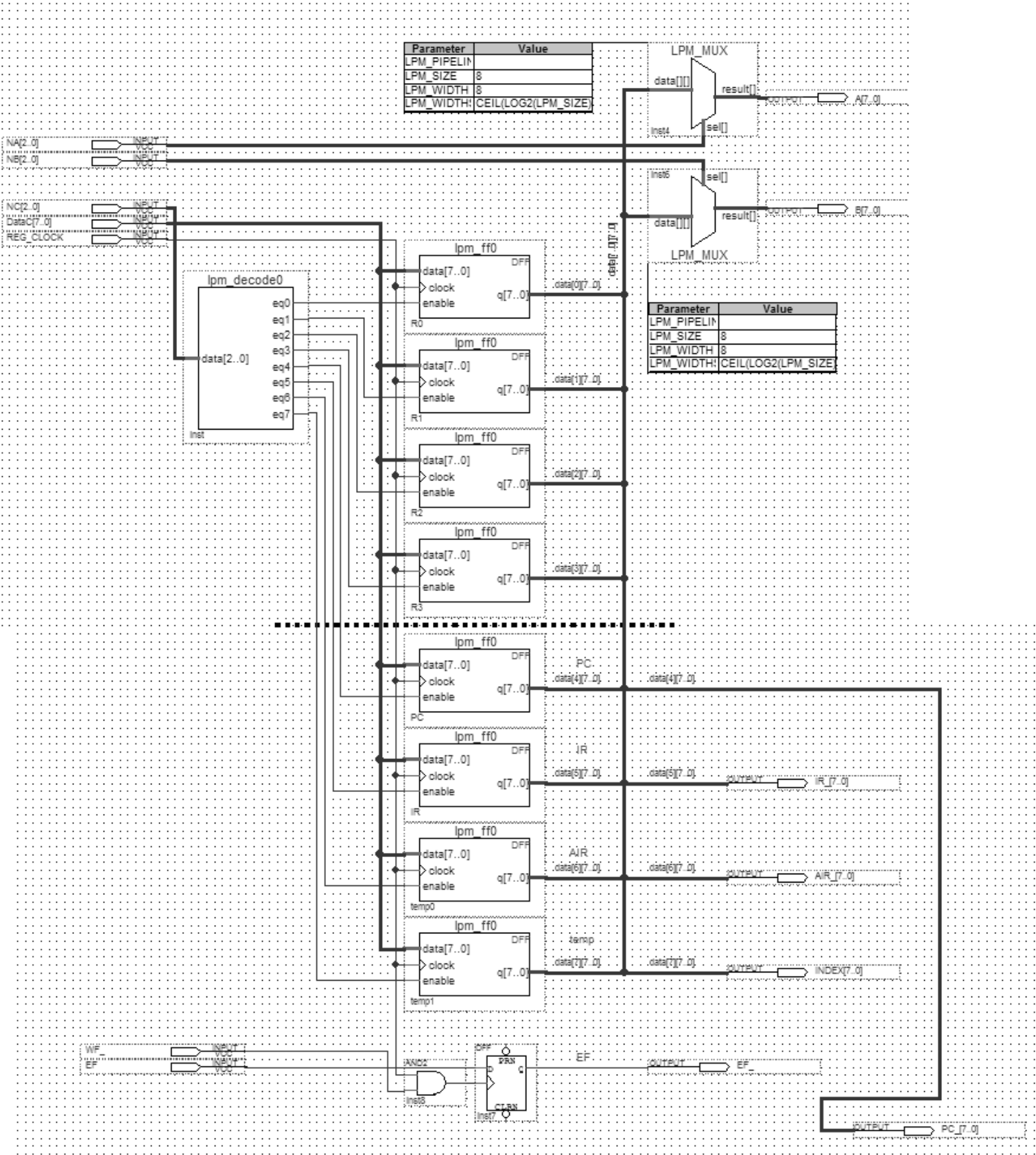
REG_SRC_SW



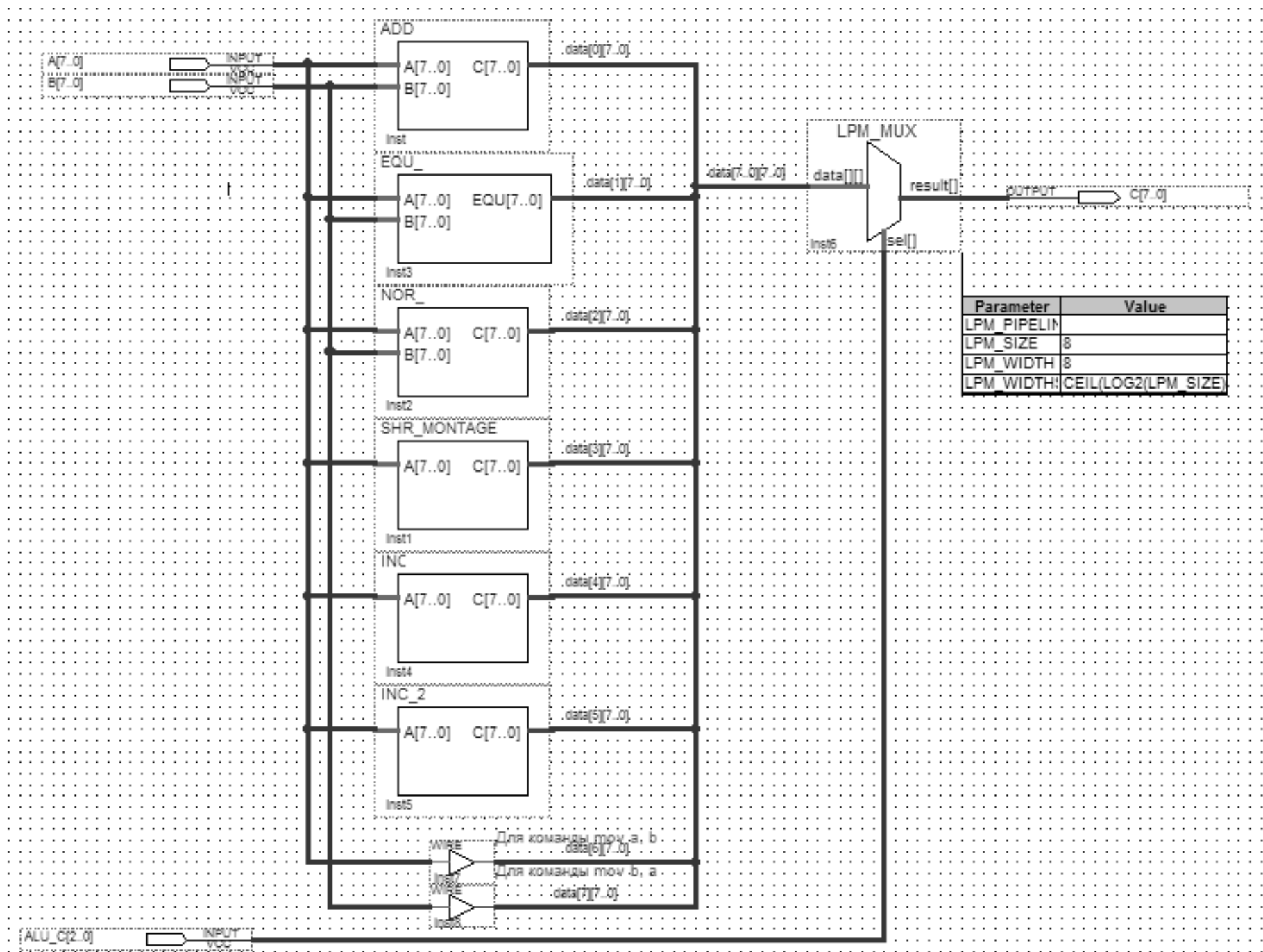
ADDR_TYPE_SW



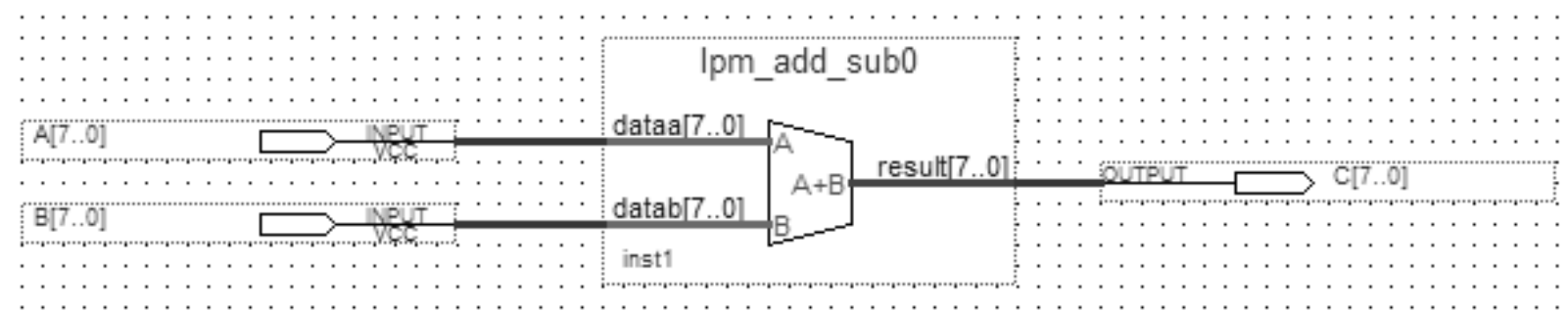
REG



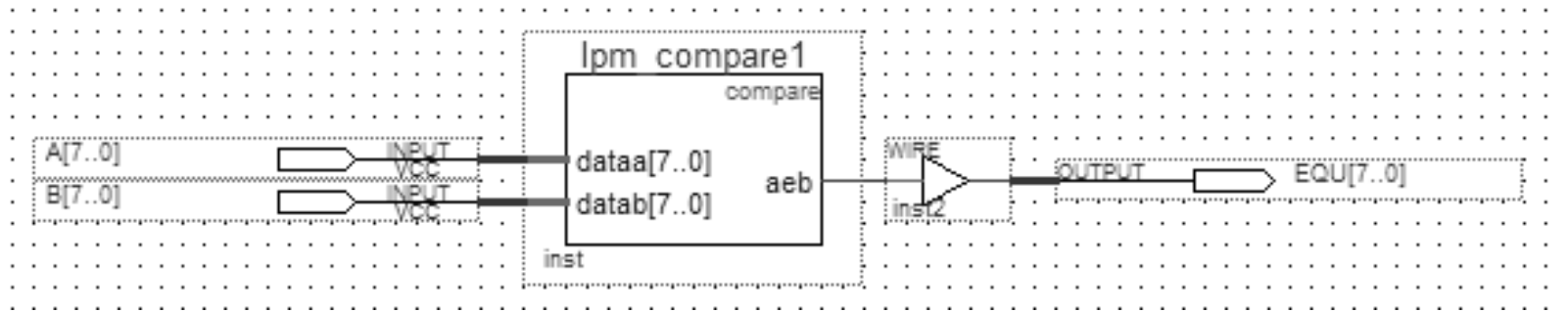
ALU



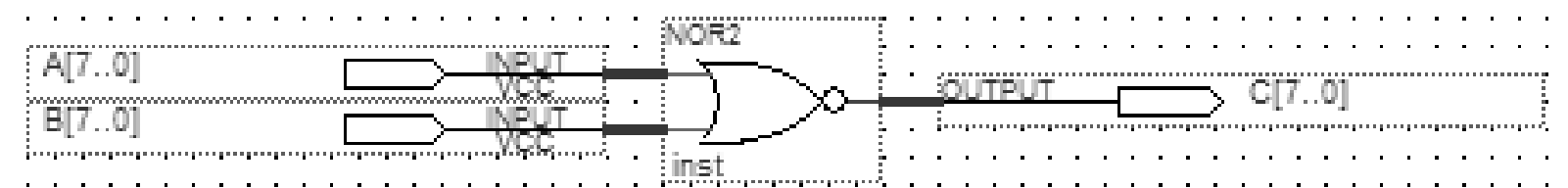
ADD



EQU



NOR



ADRESATION

