

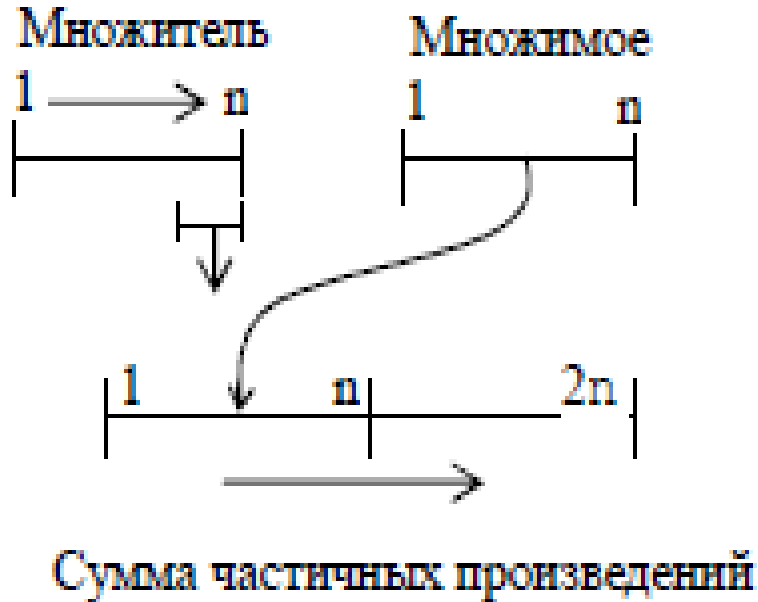
# Синтез микропрограммных управляющих автоматов

Курсовой проект

# План

- **Алгоритм умножения чисел с ФЗ первым способом в прямом коде**
- **Схема операционного автомата**
- **Содержательная ГСА**

# Алгоритм умножения чисел с ФЗ первым способом в прямом коде



## Разрядность регистров:

- Множителя и множимого –  $n$ ;
- Суммы частичных произведений –  $2n$ ;

Множимое следует прибавлять в старшие  $n$  разрядов регистра суммы частичных произведений.

## Особенность 1 способа:

- в цикле умножения возможно временное переполнение разрядной сетки мантисс (временное ПРС мантисс) в регистре суммы частичных произведений, которое ликвидируется при очередном сдвиге вправо.

## Алгоритм умножения мантисс

1. Определить знак произведения путем сложения по модулю два знаковых разрядов сомножителей.
2. Проверить множимое на равенство нулю: если равно нулю, операцию умножения следует прекратить, т.к. результат будет также равным нулю.
3. Проверить множитель на равенство нулю: если равен нулю, операцию умножения следует прекратить, т.к. результат будет также равным нулю.
4. Выполнить цикл умножения по следующим правилам:
  - Произвести анализ очередного разряда множителя.
  - Произвести суммирование множимого с суммой частичных произведений (ЧП), если цифра множителя «1», иначе перейти к п.5 алгоритма.
5. Произвести сдвиг множителя и суммы ЧП на один разряд вправо.
6. Присвоить модулю произведения знак из п.1 алгоритма.

# Схема операционного автомата

- Основная цель курсовой работы – **минимизация аппаратных затрат.**
- Чрезмерное упрощение ОА часто влечет существенное усложнение УА (и замедление его работы)

## Будем руководствоваться следующими соображениями:

- По определению первого способа регистр суммы ЧП (и, как следствие, сумматор) имеет двойную разрядность. **НО**, поскольку выходная шина имеет разрядность  $n$ , то младшие разряды произведения округляются, поэтому не имеет смысла хранить их и можем сократить разрядность суммы частичных произведений и сумматора до  $n$ .
- Так как возможно временное ПРС мантисс, то в сумматоре должен быть либо разряд переноса, либо дополнительный бит (последнее предпочтительно, т.к. экономя один бит мы не выигрываем в аппаратуре, но усложняем логику сумматора).
- Если один из операндов равен нулю, то и знак результата должен быть равен нулю, вне зависимости от знака второго операнда.

**ОА должен обеспечивать** (следует из описания алгоритма):

- Хранение операндов и суммы ЧП
- Определение знака произведения
- Проверку операндов на равенство нулю и выдачу нулевого результата, если один из сомножителей равен нулю
- Исключение возникшего временного ПРС
- Суммирование множимого и суммы ЧП
- Сдвиг множителя и суммы ЧП
- Подсчет количества циклов умножения

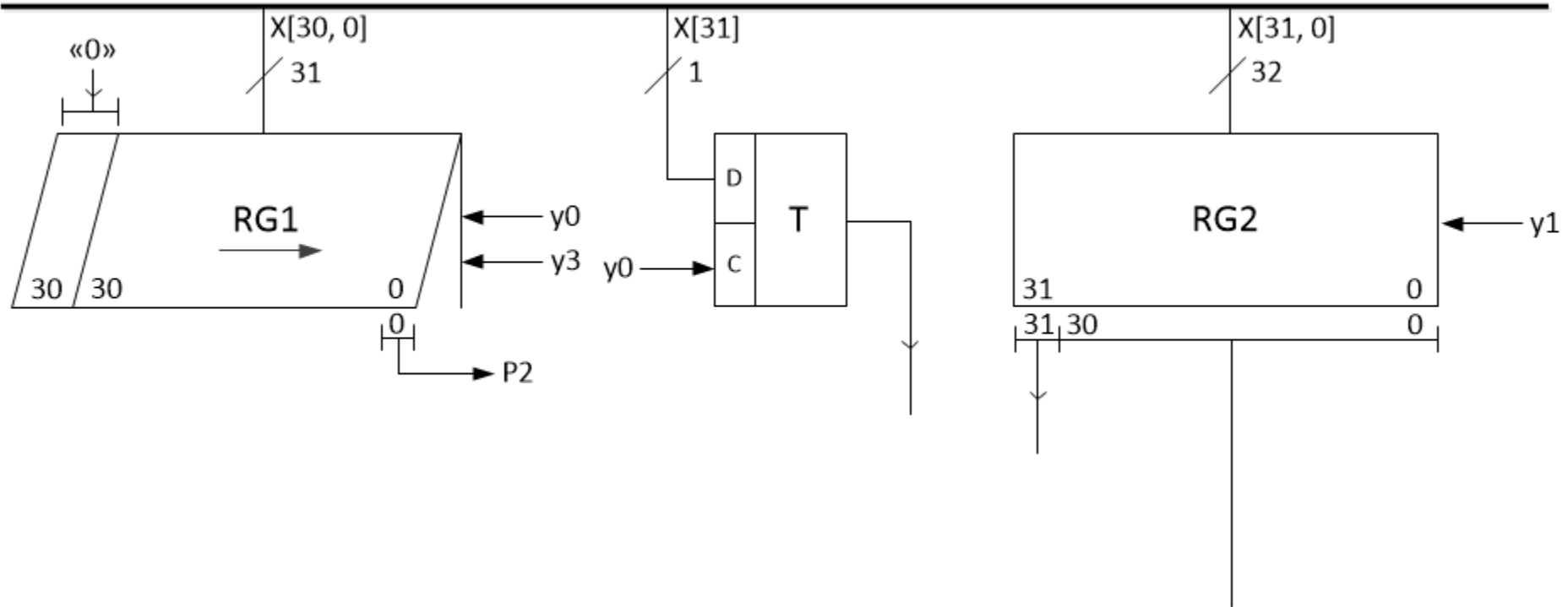
## ОА должен содержать:

- 31-разрядный сдвиговый регистр RG1 для хранения модуля множителя;
- D-триггер Т для хранения знака множителя;
- 32-разрядный регистр RG2 для хранения множимого со знаком;
- 32-разрядный регистр RG3 для хранения модуля суммы ЧП (дополнительный старший бит отводится для возможности исключения временного ПРС);
- 6-разрядный инкрементный счетчик тактов СТ;
- 32-разрядный сумматор SM;
- 31-разрядную схему «ИЛИ» с прямым и инверсным выходами для проверки операндов на равенство нулю (**одна** схема будет использоваться для проверки обоих операндов);
- Элементы «XOR» и «И» для определения знака результата;
- Усилитель-формирователь для выдачи результата на ШИВых.



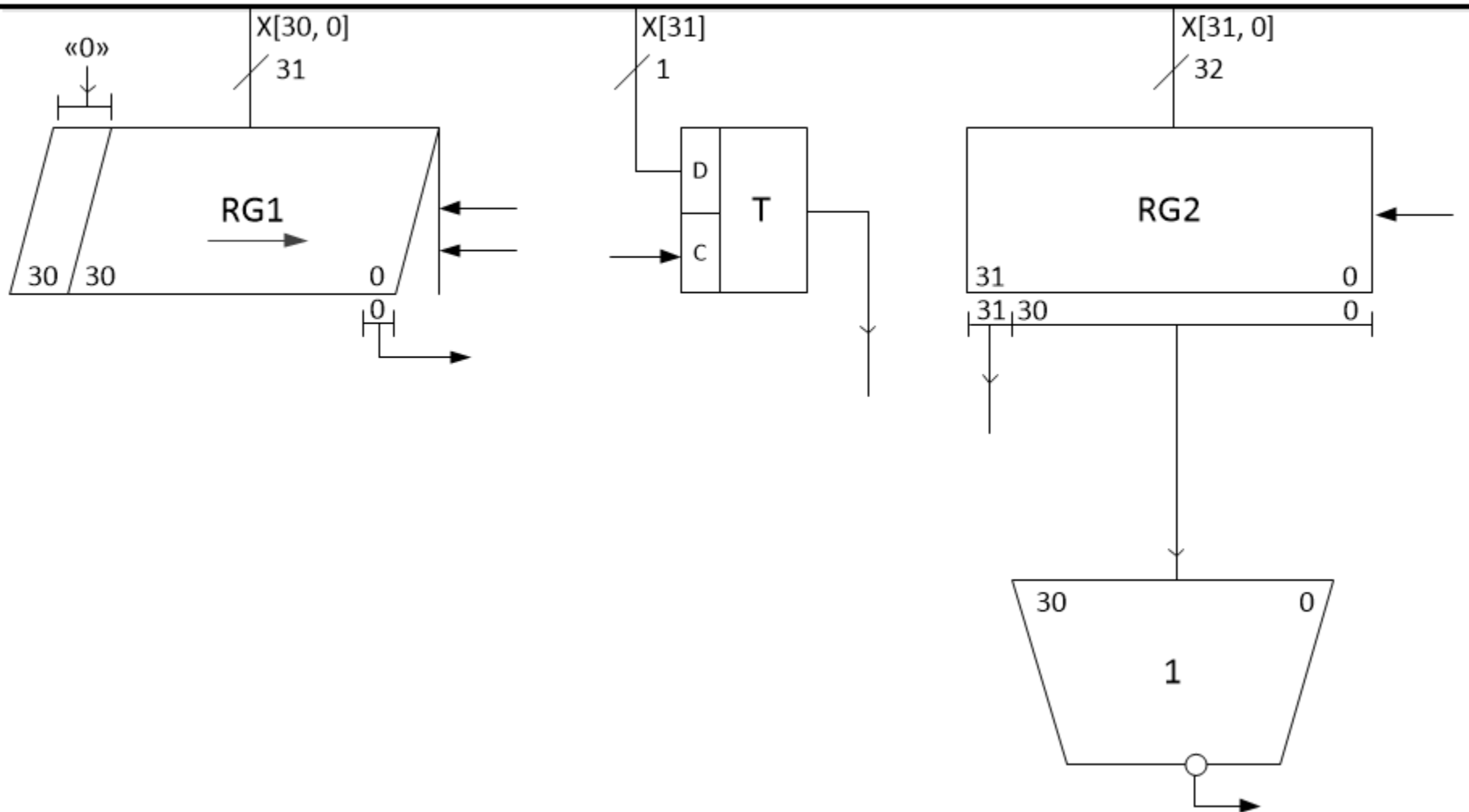
# Прием и хранение операндов:

ШИВх



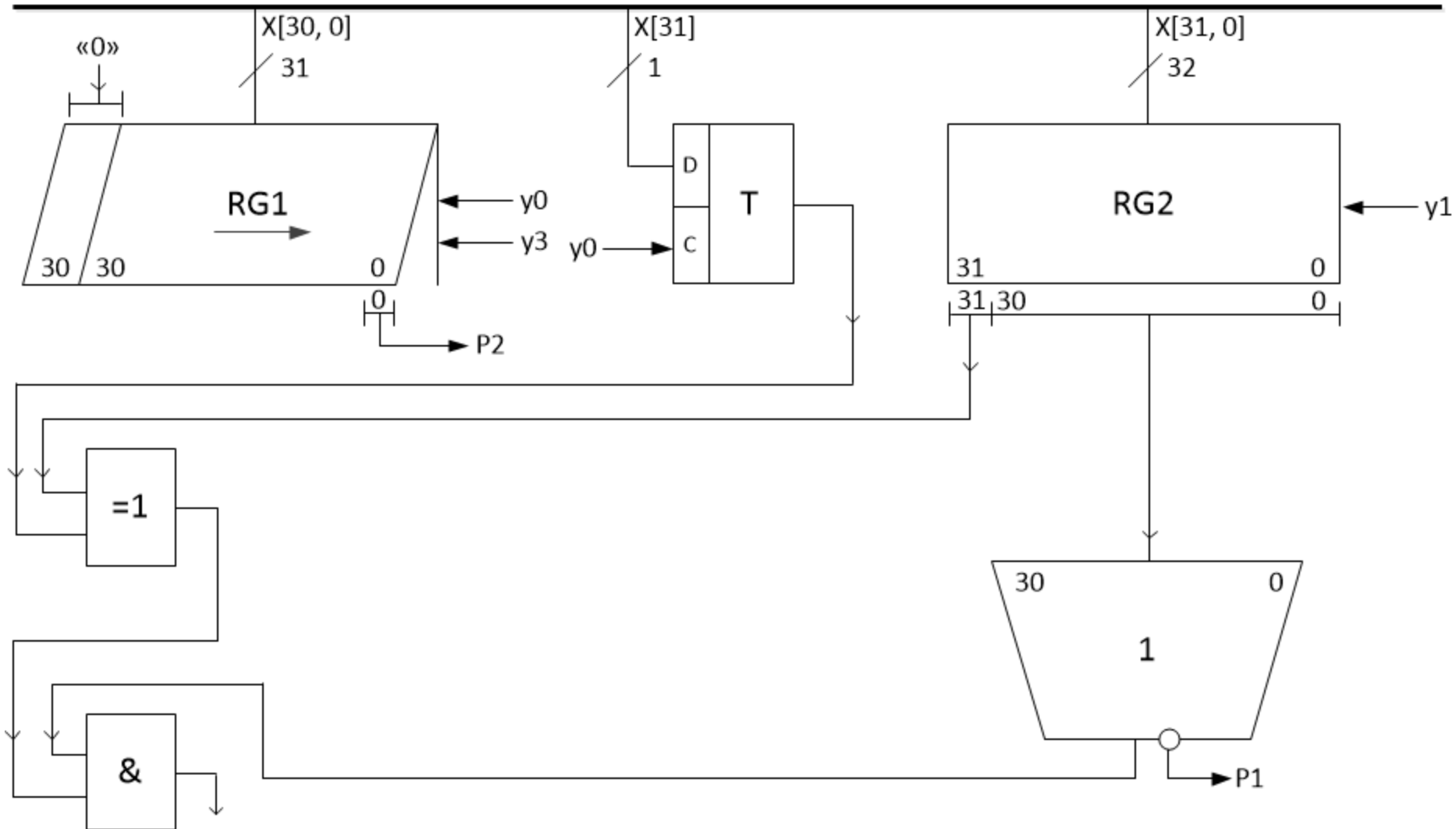
# Проверка на равенство нулю:

ШИВх



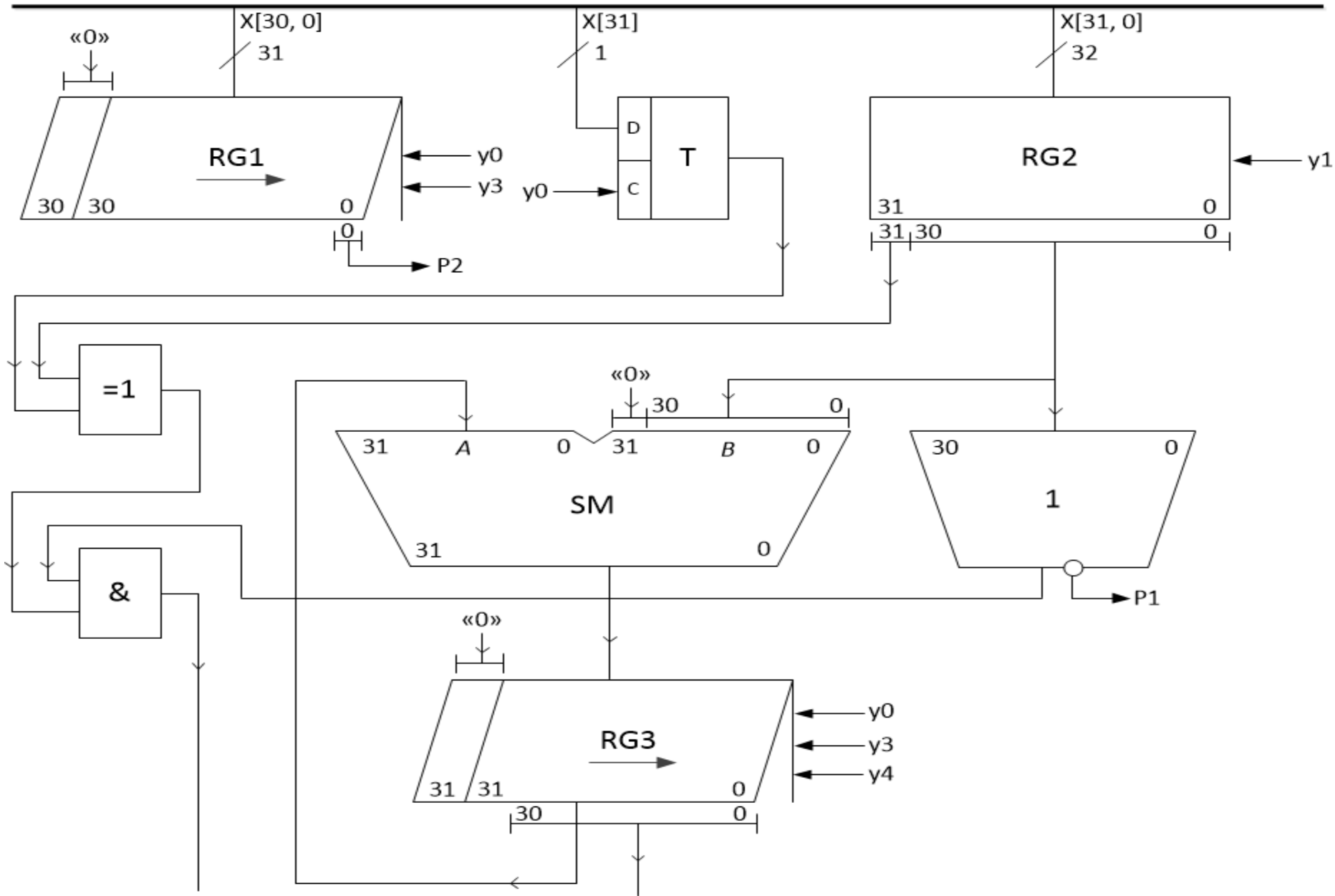
# Вычисление знака:

ШИВх



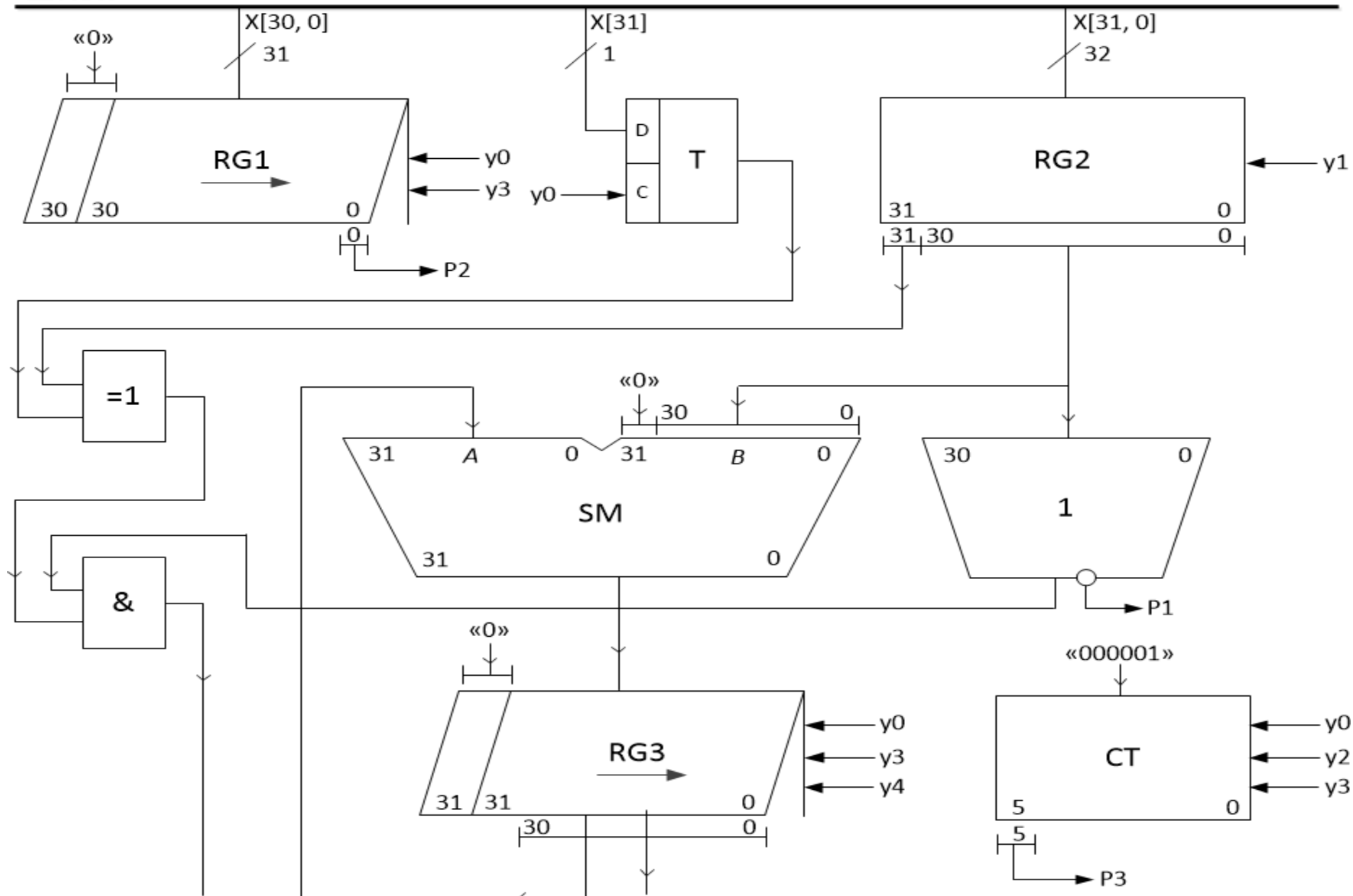
# Сложение и хранение суммы ЧП:

ШИВх



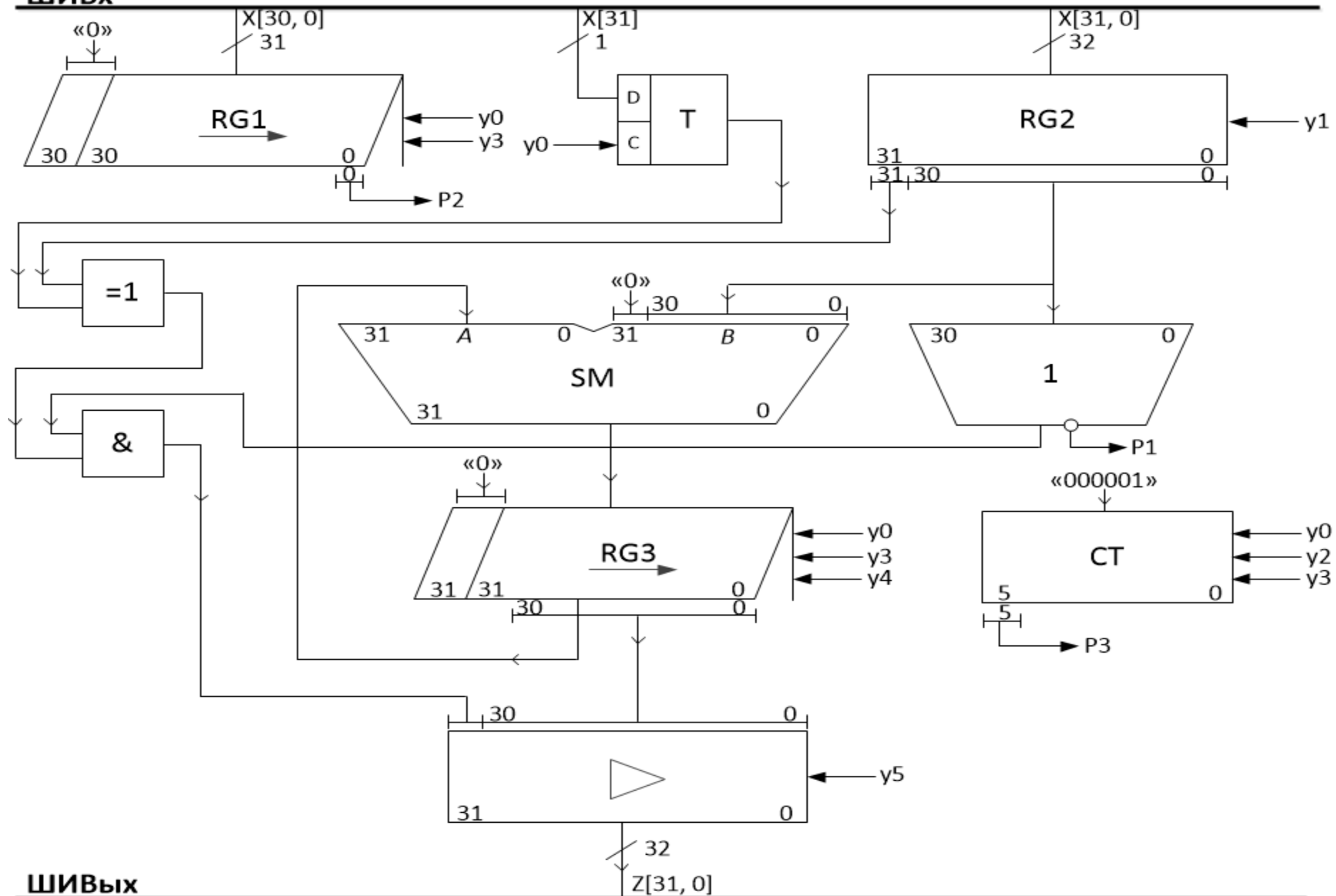
# Подсчет числа тактов:

ШИВх

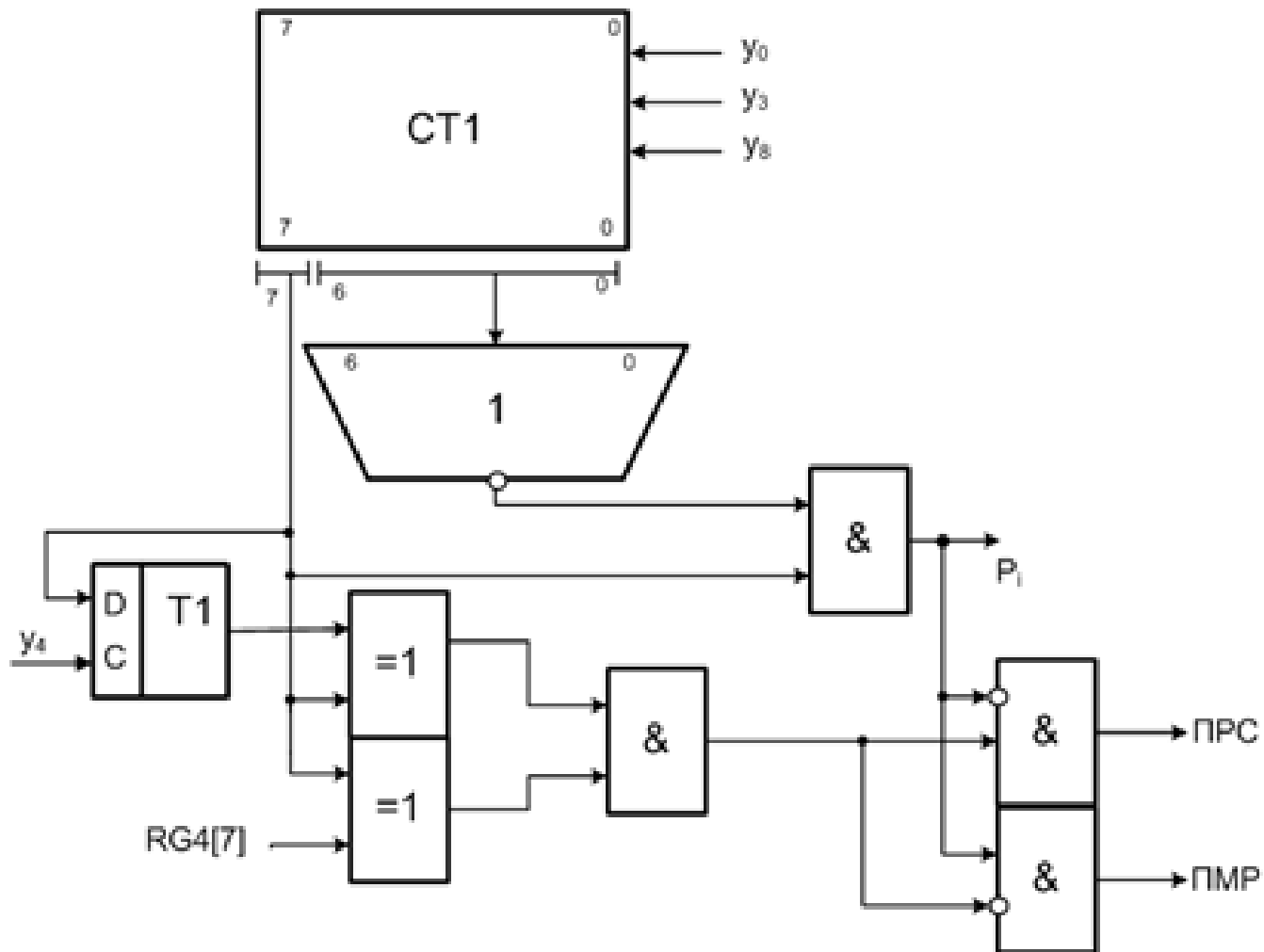


# Функциональная схема ОА:

ШИВх



## Функциональная схема ОА:



# Управляющие сигналы, поступающие из УА в ОА:

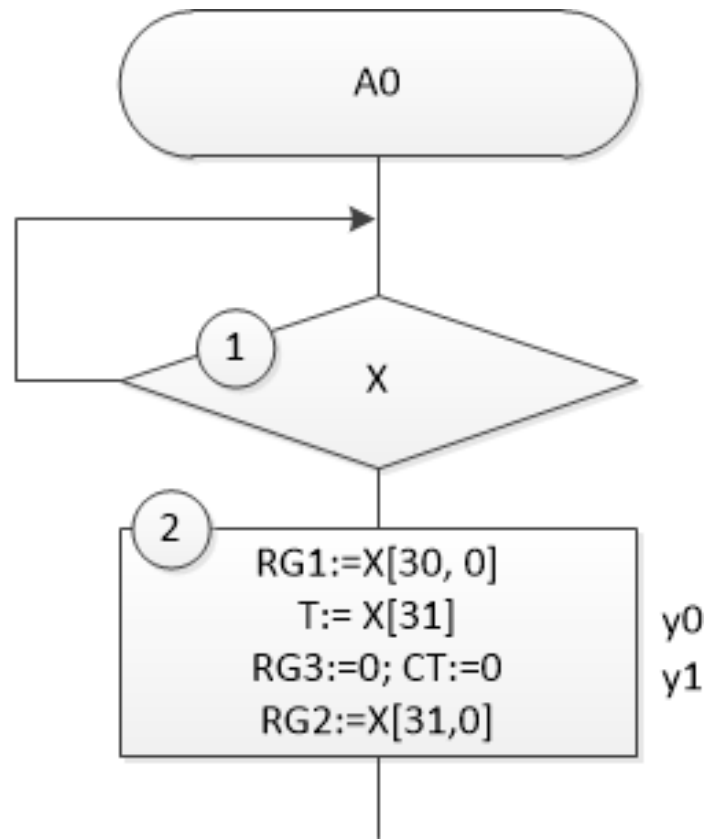
Сигнал	Микрооперация
y0	запись в RG1, запись знака множителя в T, обнуление RG3, обнуление CT
y1	запись в RG2,
y2	запись в CT
y3	сдвиг RG1 вправо $RG1 := R1(RG1)$ , сдвиг RG3 вправо $RG3 := R1(RG3)$ , $CT = CT + 1$
y4	запись в RG3
y5	выдача результата



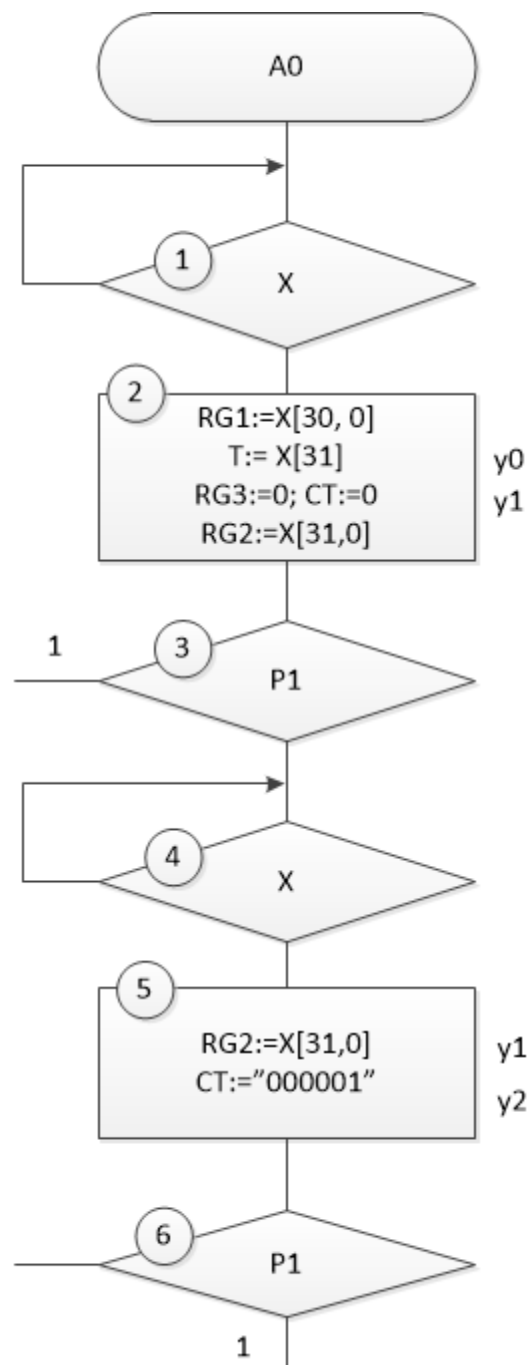
# Осведомительные сигналы, поступающие из ОА в УА:

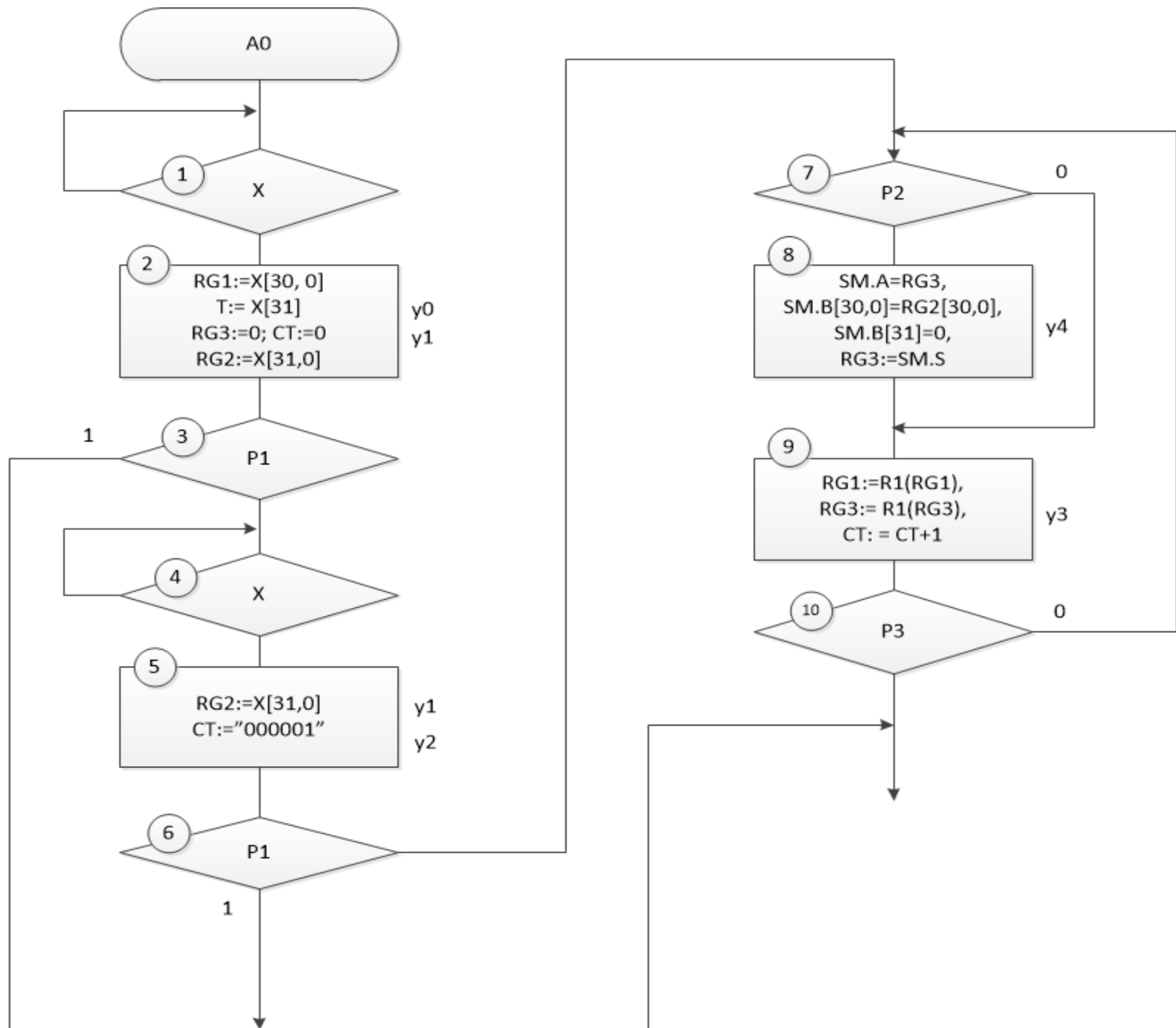
Сигнал	Логическое условие (состояние ОА)
X	проверка наличия операндов на ШИВх
P1 = 1	один из операндов равен нулю
P2 = 1	необходимо выполнять сложение суммы ЧП и множимого
P3 = 1	цикл умножения завершен
Z	проверка возможности выдачи результата на ШИВых

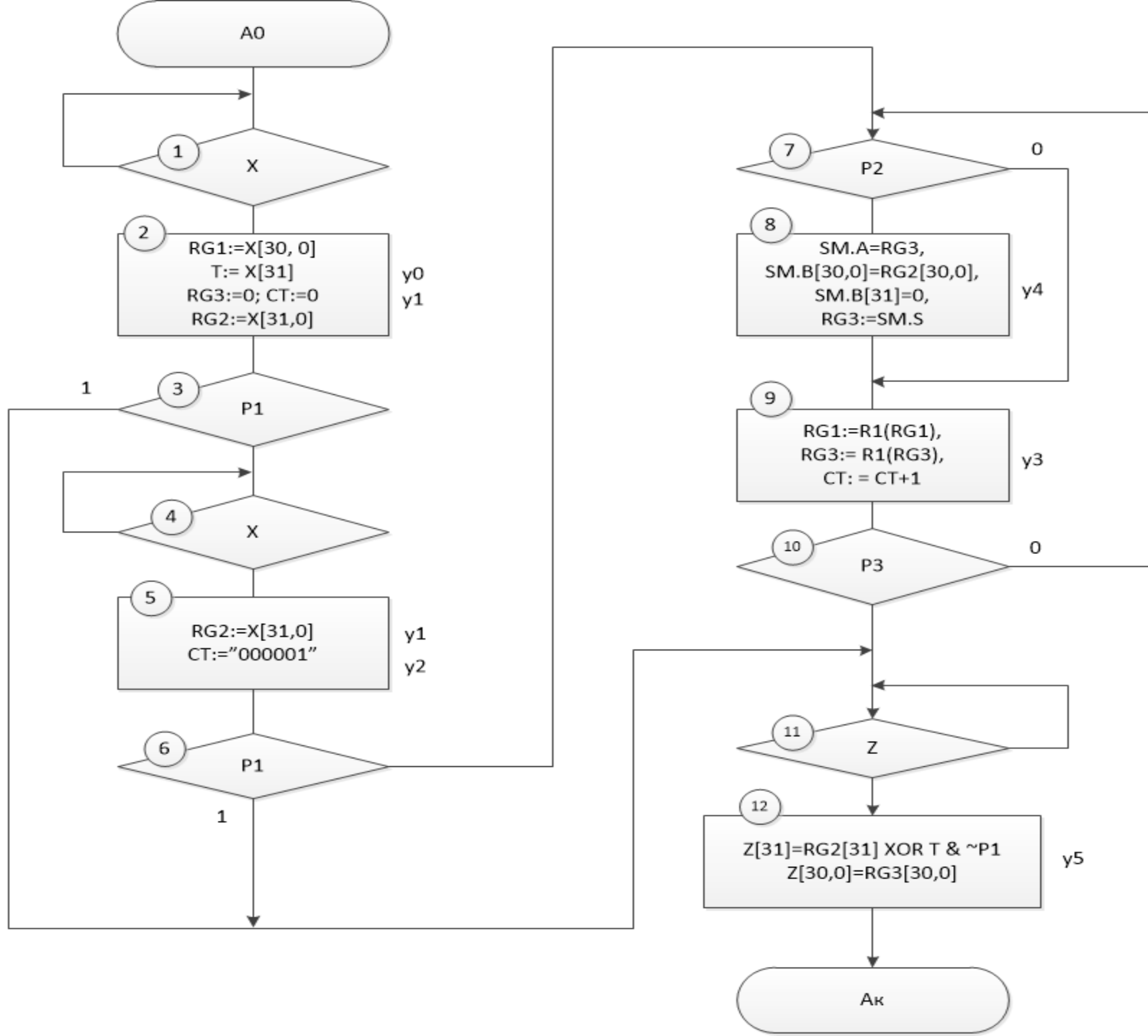
# Содержательная ГСА



Обратить внимание на знак «:=» - он применяется для обозначения **записи** в элемент памяти







# На что следует обратить внимание

- Процесс проектирования творческий
- Каждое ЛУ ведет к усложнению структуры УА, **однако** не стоит избавляться от ЛУ всеми возможными способами, строя громоздкие комбинационные схемы в ОА
- Процесс проектирования итерационный
- Не следует забывать про основную цель проекта
- Грубыми ошибками считается как отсутствие необходимых схем проверки (ПРС, ПМР, равенство нулю и пр.), так и явно излишняя аппаратура
- Функциональная схема ОА и ГСА проектируются одновременно

# CAPIR Quartus

