Лекция 17

7 апреля

Распределение слагаемых

```
section .data
 x dq  1e200
 y dq  1e200
 z dq  1e200

section .bss
 r resq 1
 cw resw 1

section .rodata
 lc db '%lf', 10, 0

CMAIN:
   ; … пролог функции
   sub     esp, 20
```

```
fstcw   word [cw]
and  word [cw], 11111111_11000000b
fldcw   word [cw]

fld     qword [x]
fld     qword [z]
fmulp
fld     qword [y]
fld     qword [z]
fmulp
fsubp
; вызов printf

add     esp, 20
; … эпилог функции
```

Польская обратная запись

- `(w + x + y + z) / 4`
- `w x + y + z + 4 /`

```
section .data
 w dq  1e10
 x dq  1e10
 y dq  1e10
 z dq  1e10
 d dd  4
```

```
section .text
    ;…
    fld     qword [w]
    fld     qword [x]
    faddp
    fld     qword [y]
    faddp
    fld     qword [z]
    faddp
    fild    dword [d]
    fdivp
    ;…
```

Предопределенные константы

- На «верхушку» стека регистров (ST0) помещается определенная константа
  - FLD1 +1.0
  - FLDL2T $\log_2 10$
  - L2E $\log_2 e$
  - FLDPI π
  - FLDLG2 $\log_{10} 2$
  - FLDLN2 $\log_e 2$
  - FLDZ +0.0

Некоторые команды x87

- `fld`
- `fstp / fst`

Загрузка/выгрузка числа с плавающей точкой

- `fild`
- `fistp / fist`

Загрузка/выгрузка целых чисел с конвертацией

- `fchs`
- `faddp`
- `fsubp / fsubrp`
- `fmulp`
- `fdivp / fdivrp`

Арифметика

- fsqrt
- fcos / fsin
- fsincos
- fptan
- f2xm1

Арифметика+ тригонометрия

- ftst
- fucom / fucomp / fucompp

Сравнение чисел

- fnstsw

Выгрузка слова состояния

Сравнение чисел

```
_Bool isLe(double x, float y) {
   return x <= y;
}
```

Результат сравнения	C3	C2	C0
St0 > St1	0	0	0
St0 < St1	0	0	1
St0 == St1	1	0	0
неопределенно	1	1	1

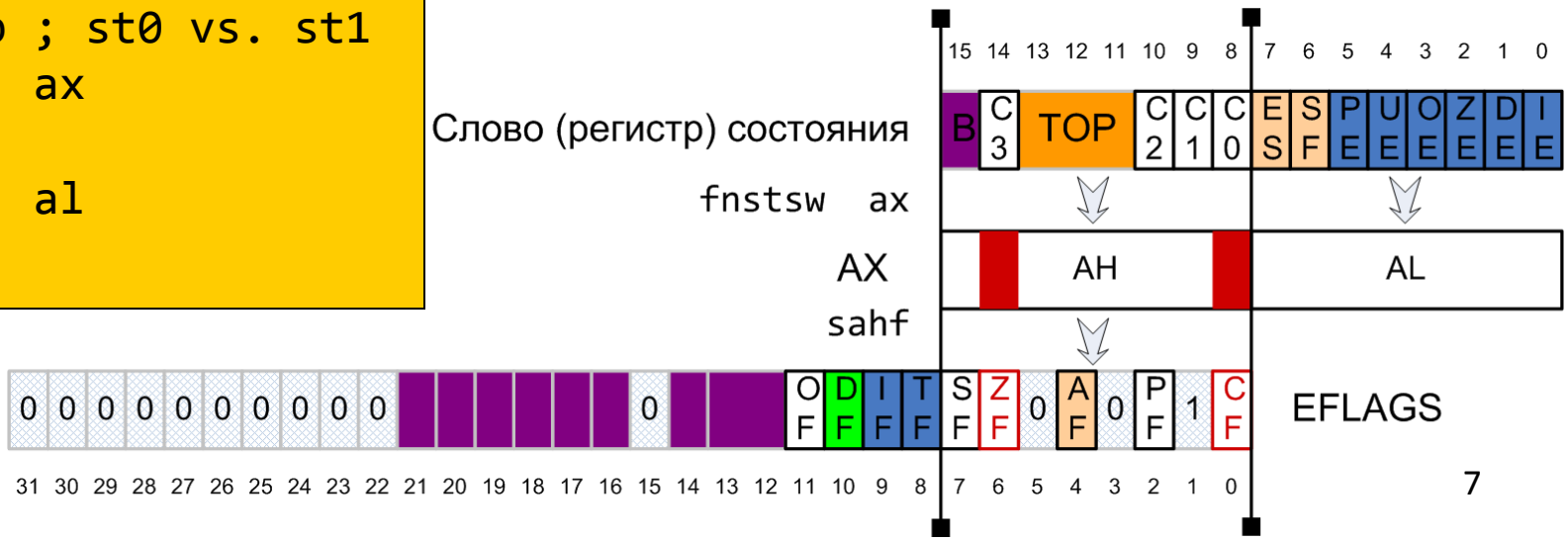
```
isLe:
        push    ebp
        mov     ebp, esp
        fld     dword [ebp+16]
        fld     qword [ebp+8]
        fucompp ; st0 vs. st1
        fnstsw  ax
        sahf
        setbe   al
        pop     ebp
        ret
```

Извлечение результатов сравнения

- C3 → ZF, C0 → CF
- Можно использовать условные коды, применяемые при сравнении беззнаковых чисел

Результат сравнения	C3	C2	C0
ST(0) > ST(i)	0	0	0
ST(0) < ST(i)	0	0	1
ST(0) == ST(i)	1	0	0
неупорядочены	1	1	1

```
isLe:
    ...
    fld     dword [ebp+16]
    fld     qword [ebp+8]
    fucompp ; st0 vs. st1
    fnstsw  ax
    sahf
    setbe   al
    ...
```



Функции: возвращение числа с плавающей точкой

```
void caller(double *p) {
   *p = inverse(*p);
}

caller:
   push    ebp
   mov     ebp, esp
   sub     esp, 8
   mov     eax, dword [ebp+8]
   fld     qword [eax]
   fstp    qword [esp]
   call    inverse
   mov     eax, dword [ebp+8]
   fstp    qword [eax]
   leave
   ret
```

```
float inverse(double x) {
   return 1/x;
}

inverse:
   push    ebp
   mov     ebp, esp
   fld1
   fld     qword [ebp+8]
   fdivp
   pop ebp
   ret
```

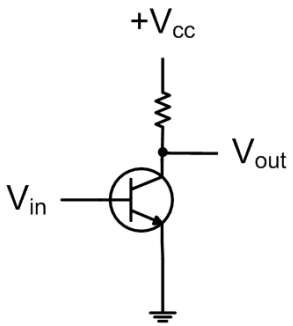
Итоги второй части

- Процессор с точки зрения пользовательской программы
- Основные механизмы реализации языка Си
  - Соответствие типов
  - Указатели, адресная арифметика
  - Управляющие операторы: разные стратегии реализации
  - Размещение данных в памяти
    - Классы памяти: статическая, автоматическая, динамическая
    - Выравнивание
  - Соглашения о вызове функции
- ABI – Application Binary Interface
- IA-32: x86 + x87 + ...
  - Некоторые особенности устройства

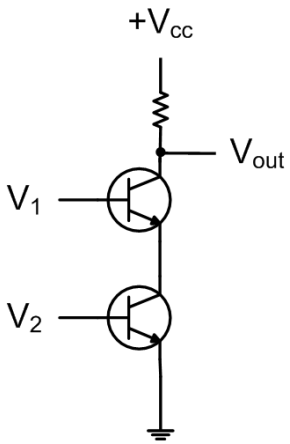
Далее: Архитектура ЭВМ

- Организация аппаратного обеспечения компьютера
  - Физические основы, шины, периферийные устройства
- Организация памяти
  - НЖМД, кэш
  - Производительность
- Организация ЦПУ
  - Конвейер, система команд
- Многозадачная работа компьютера
  - Изоляция пользовательских программ

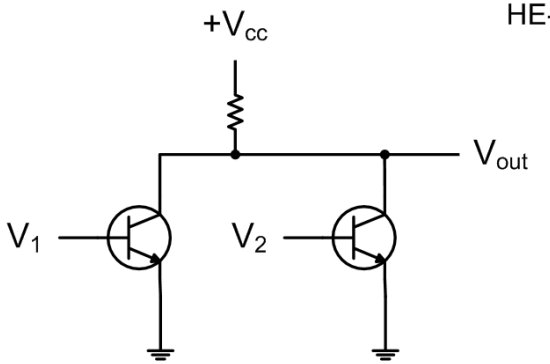
Логические вентили



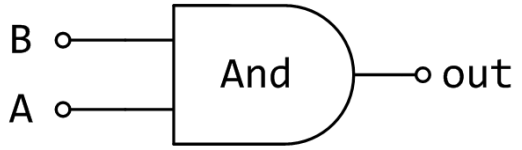
Инвертор

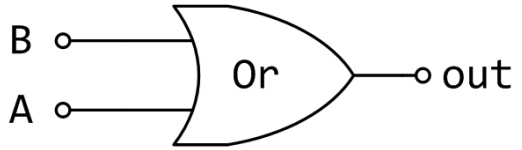


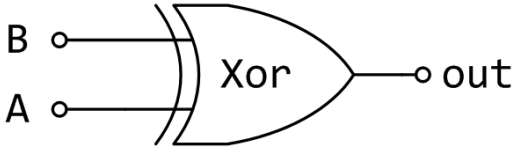
НЕ-И

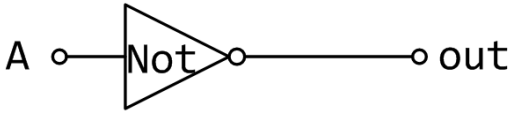


НЕ-ИЛИ

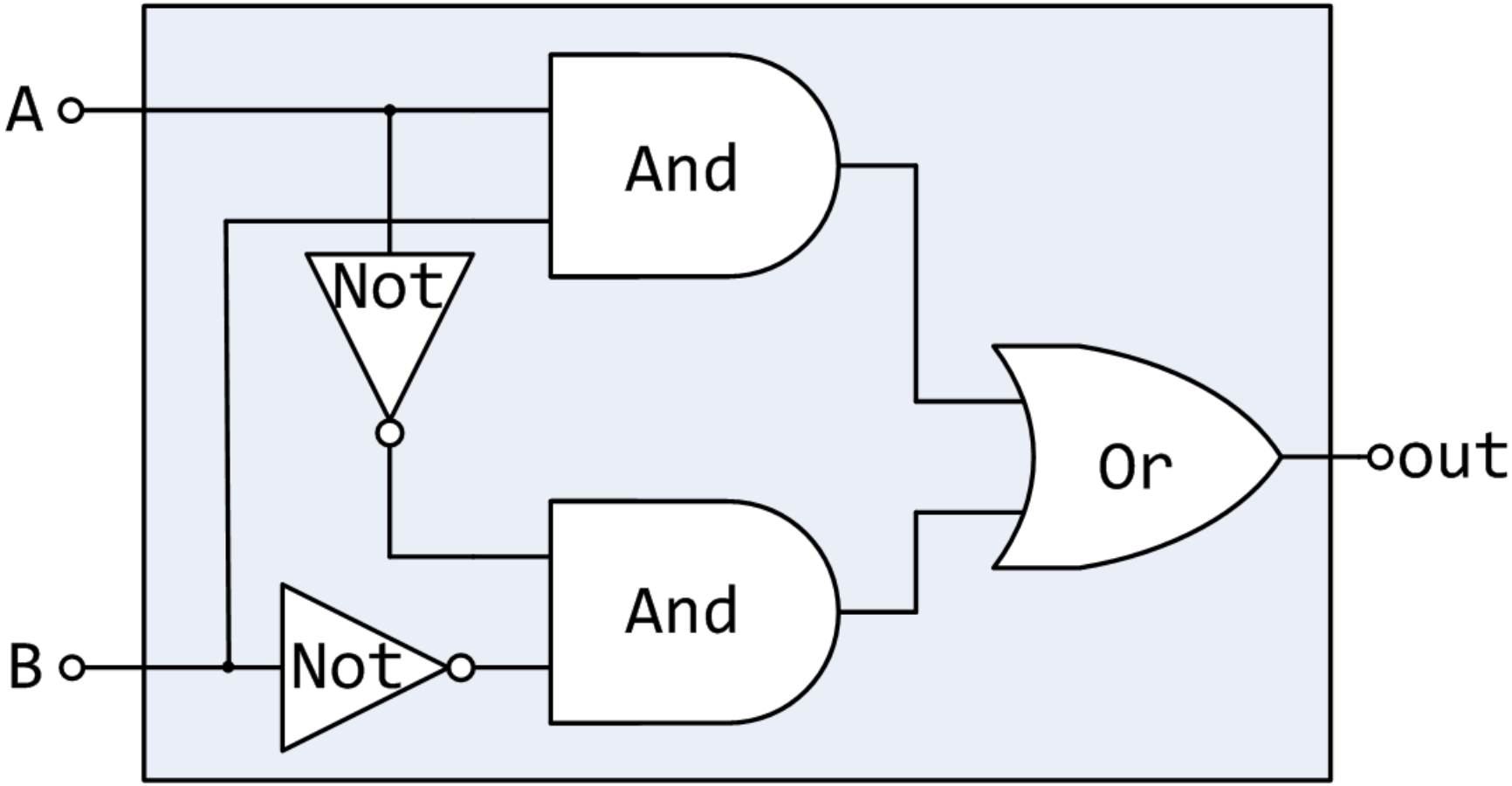




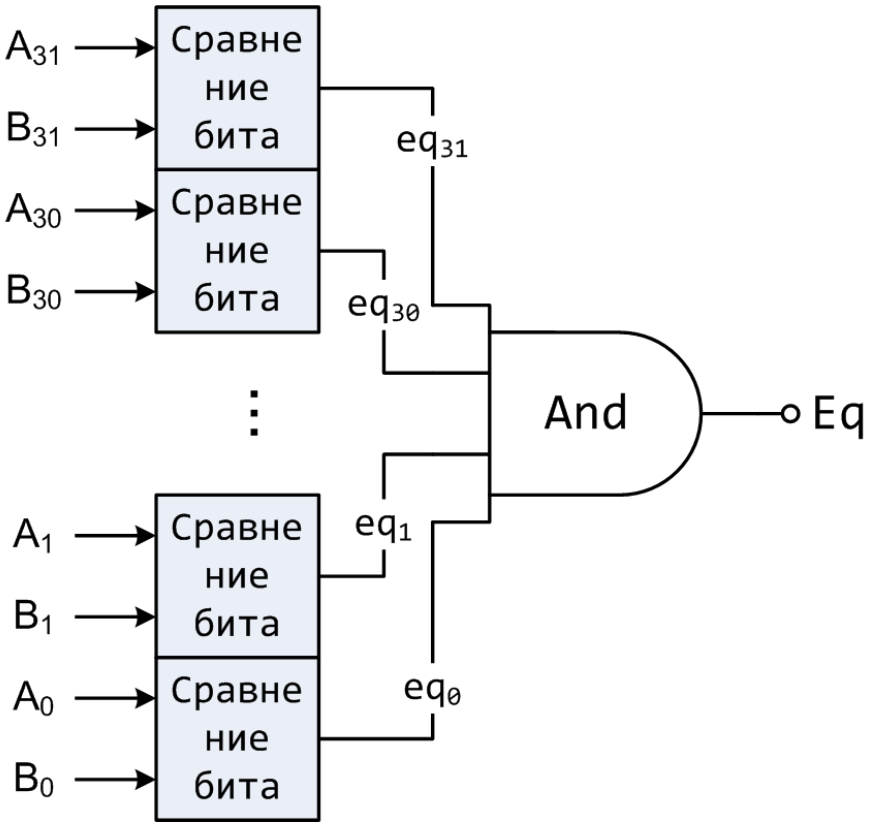


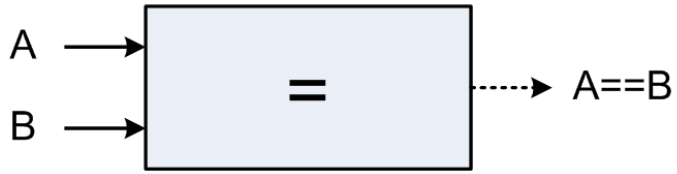


Сравнение битов



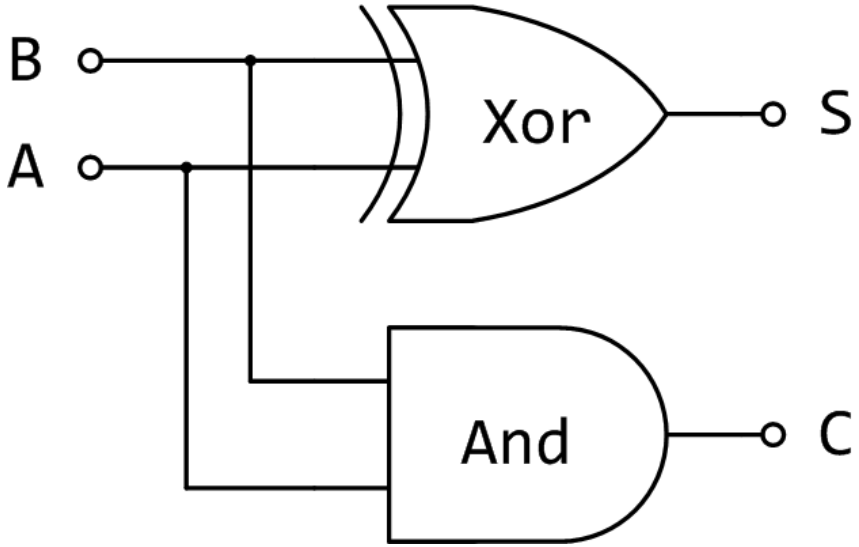
Сравнение слов



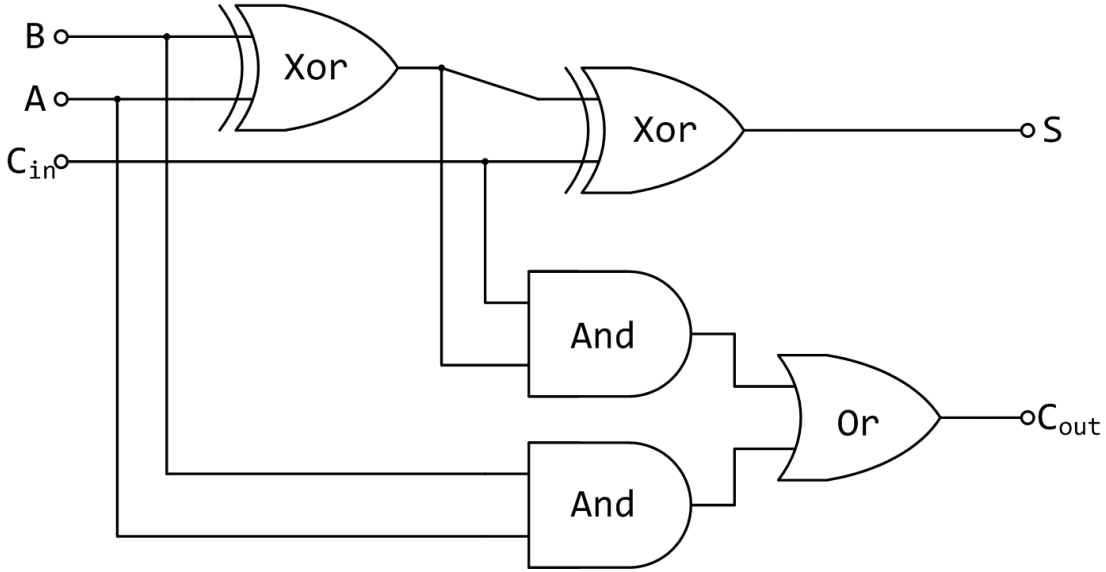


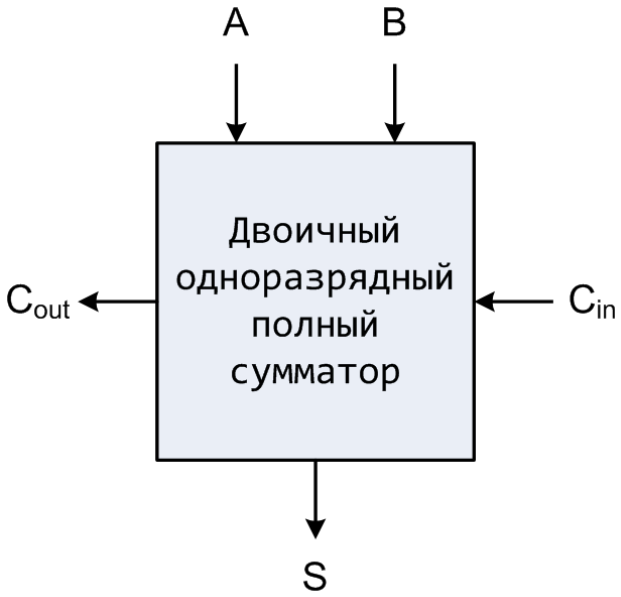
Полусумматор

- Суммирует биты A и B
- Результат – бит S
- Перенос – бит C

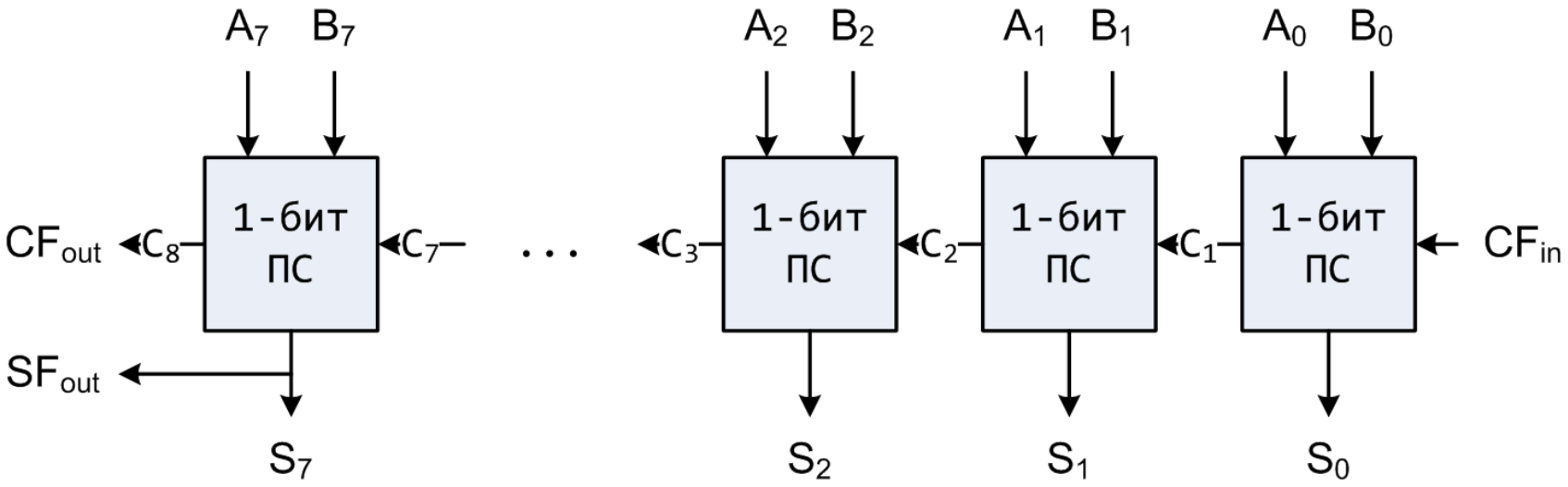


Полный двоичный сумматор

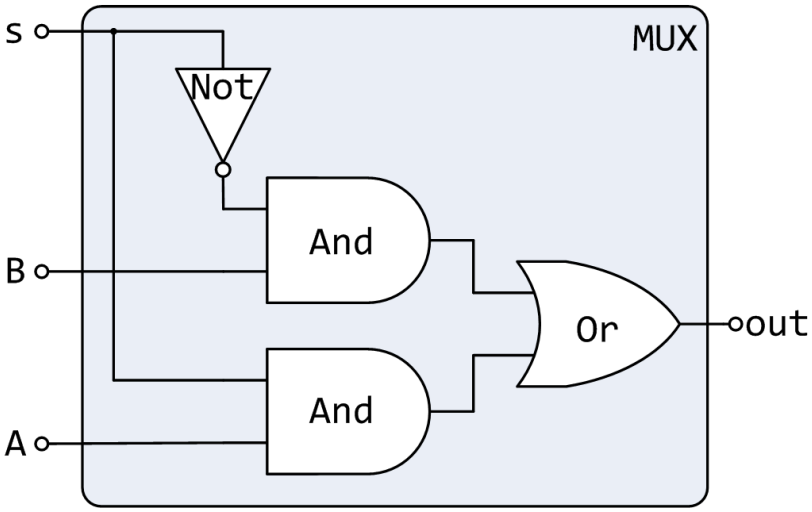


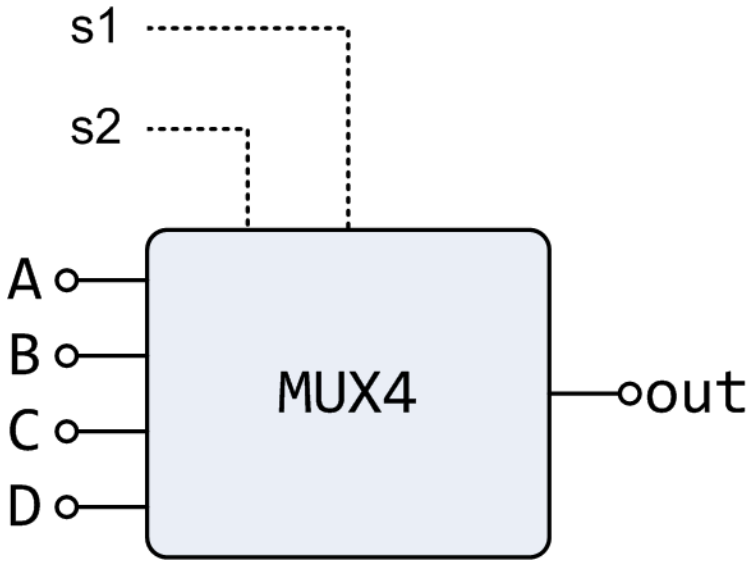


Суммирование слов

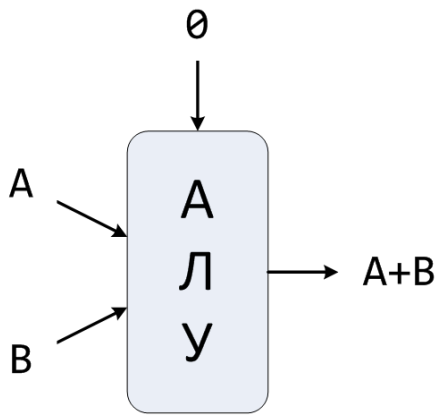


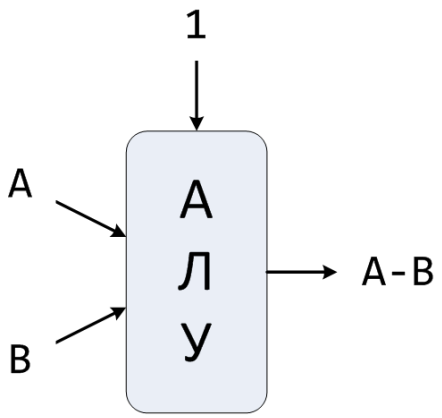
Мультиплексор

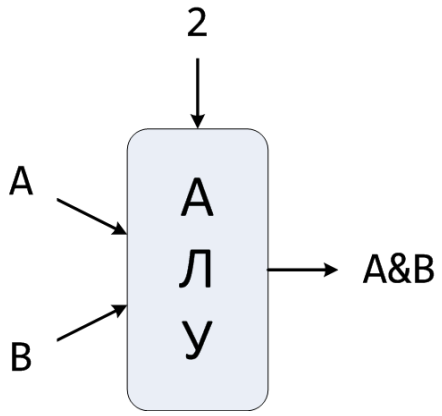


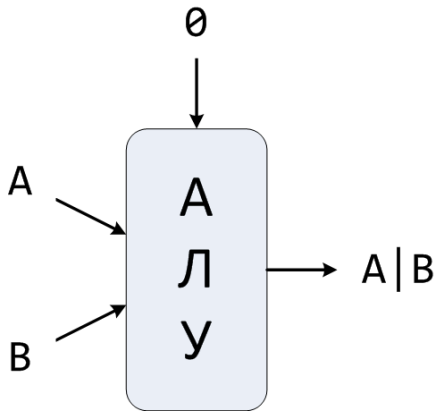


АЛУ

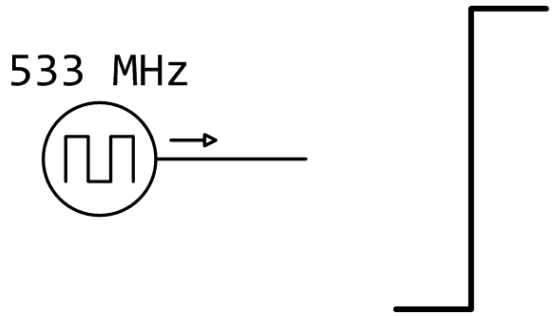




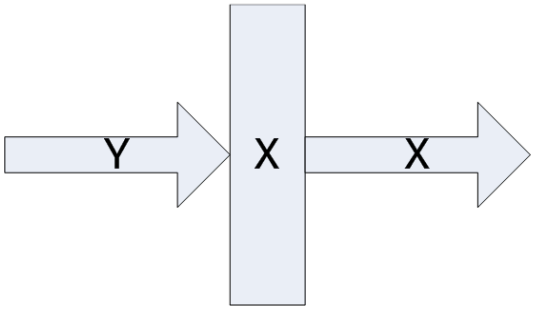




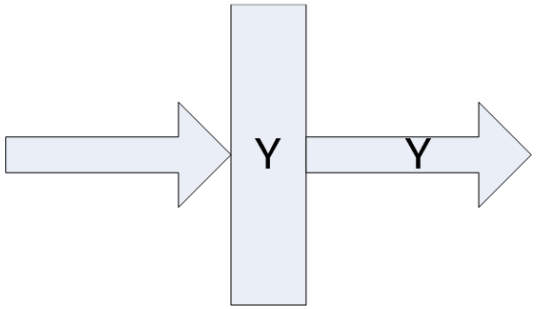
Регистр: сохранение 1 бита

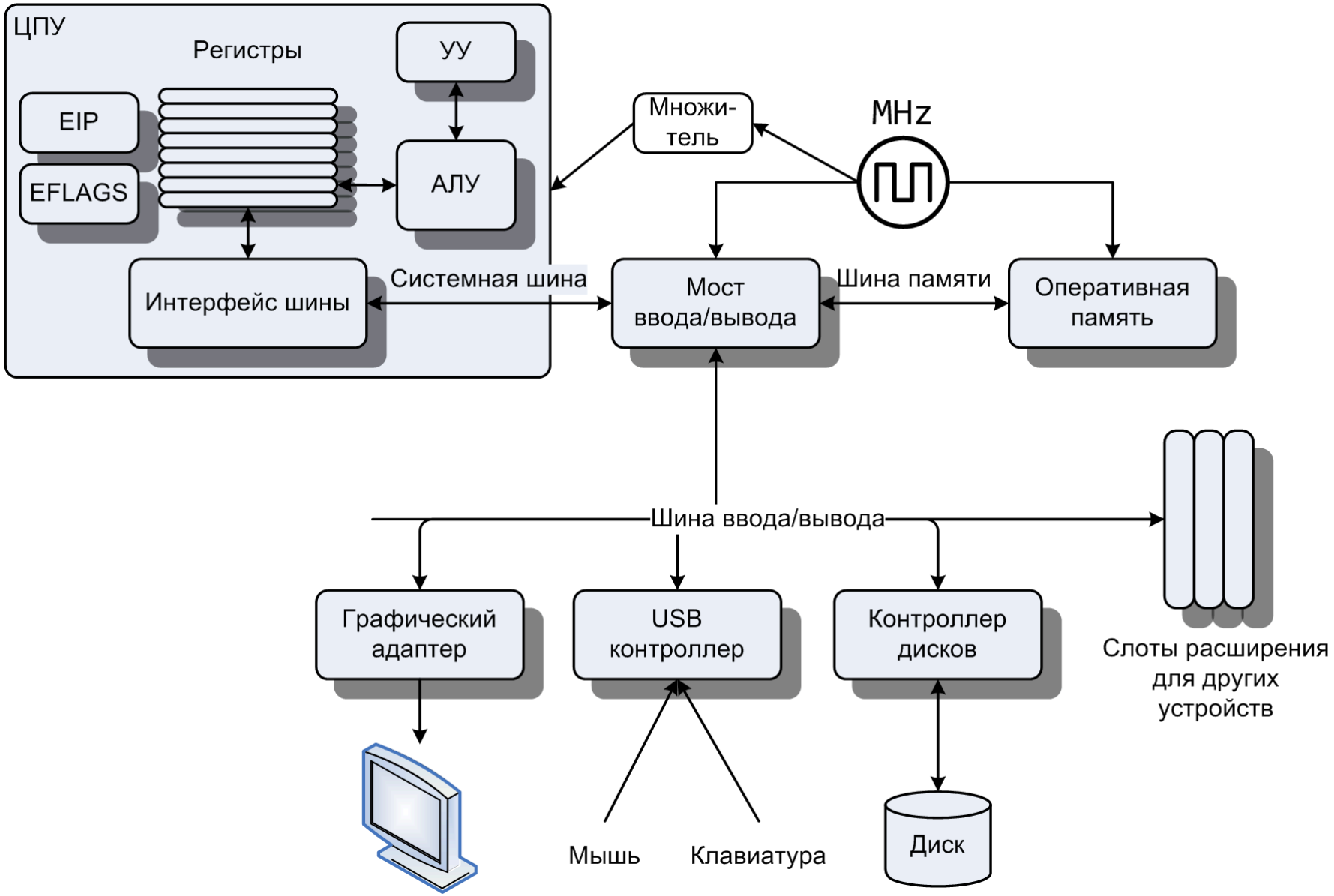


Состояние регистра = X



Состояние регистра = Y





Закон Мура (Moore's law)

- Число транзисторов на кристалле будет удваиваться каждые 24 месяца
- Гипотеза выдвинута в 1965 году Гордоном Муром (один из основателей Intel)
- Ограничения
  - Атомарная природа вещества
  - Скорость света
- Негативная сторона – предельно быстрое устаревание вычислительной техники
- Открытый вопрос: область применимости

Закон Гроша (Grosch's law)

- Производительность компьютера увеличивается как квадрат стоимости
  - Емкость мирового рынка компьютеров – 5 машин
- Гипотеза выдвинута в 1965 году Хербом Грошем (второй ведущий ученый IBM, после Эккерта)
- 1997: закон полностью опровергнут
- Применимость к определенному классу машин
  - Рабочая станция
  - Майнфрейм
  - Суперкомпьютер
- Новые вычислительные/информационные ресурсы
  - Поисковые системы
  - Облачные вычисления

Коллоквиум

- Первая половина лекции 11 апреля
- Длительность – 50 минут
- 6 задач
  - (1) Вычислить размер типа данных, определить смещения полей
    - Массивы, структуры, объединения
    - Выравнивание: Linux/Windows
  - Функции
    - (2) Восстановить прототип функции, включая порядок параметров
    - (3) Реализовать функцию с заданным соглашением / omit frame pointer
    - (4) Выравнивание фрейма, вызов библиотечной функции
  - (5) Строковые команды
  - (6) Представление чисел с плавающей точкой
- Каждая задача – от 0 до 6 баллов

Оценка	**2**	**3**	**4**	**5**
%	[0, 40)	[40, 60)	[60, 80)	[80, 100]