|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **ИУ**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА **ИУ1** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ОТЧЕТ ПО ДОМАШНЕМУ ЗАДАНИЮ**

Студент\_\_\_Леончик\_Тимофей\_Сергеевич\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*фамилия, имя, отчество*

Группа\_\_\_\_\_\_\_ИУ1-11Б\_\_\_\_\_\_\_\_\_

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

*подпись, дата фамилия, и.о.*

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

*подпись, дата фамилия, и.о.*

Балл \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*2025 г.*

**Вариант 2**

**Задание №1**

1. Перевод двоичного числа 1011011101100010.0101₂ в десятичную систему:

1011011101100010₂ = 1×2¹⁵ + 0×2¹⁴ + 1×2¹³ + 1×2¹² + 0×2¹¹ + 1×2¹⁰ + 1×2⁹ + 1×2⁸ + 0×2⁷ + 1×2⁶ + 1×2⁵ + 0×2⁴ + 0×2³ + 0×2² + 1×2¹ + 0×2⁰ = 32768 + 8192 + 4096 + 1024 + 512 + 256 + 64 + 32 + 2 = 46946₁₀

0.0101₂ = 0×2⁻¹ + 1×2⁻² + 0×2⁻³ + 1×2⁻⁴ = 0.25 + 0.0625 = 0.3125₁₀

Результат: 1011011101100010.0101₂ = 46946.3125₁₀

2. Перевод через шестнадцатеричную систему

Целая часть 46946₁₀ :

46946 ÷ 16 = 2934 (ост. 2)

2934 ÷ 16 = 183 (ост. 6)

183 ÷ 16 = 11 (ост. 7)

11 ÷ 16 = 0 (ост. 11 = B)

B762₁₆

Дробная часть 0.3125₁₀ :

0.3125 × 16 = 5.0 → 5

0.5₁₆

Результат: B762.5₁₆

Проверка :

B₁₆ = 1011₂

7₁₆ = 0111₂

6₁₆ = 0110₂

2₁₆ = 0010₂

.5₁₆ = .0101₂

1011011101100010.0101₂ = 1011011101100010.0101₂

3. Перевод через восьмеричную систему

1. Целая часть 46946₁₀

46946 ÷ 8 = 5868 (ост. 2)

5868 ÷ 8 = 733 (ост. 4)

733 ÷ 8 = 91 (ост. 5)

91 ÷ 8 = 11 (ост. 3)

11 ÷ 8 = 1 (ост. 3)

1 ÷ 8 = 0 (ост. 1)

Читаем остатки снизу вверх: 133542₈

2. Дробная часть 0.3125₁₀

0.3125 × 8 = 2.5 (целая часть 2)

0.5 × 8 = 4.0 (целая часть 4)

0.24₈

Результат: 133542.24₈

Проверка :

133542₈ = 1×8⁵ + 3×8⁴ + 3×8³ + 5×8² + 4×8¹ + 2×8⁰

= 32768 + 12288 + 1536 + 320 + 32 + 2 = 46946₁₀

0.24₈ = 2×8⁻¹ + 4×8⁻² = 0.25 + 0.0625 = 0.3125₁₀

**Задание №2**

*1. Перевод 9BC7.3D₁₆ в двоичную систему*

9BC7.3D₁₆ = 1001 1011 1100 0111. 0011 1101₂

Результат: 1001101111000111.00111101₂

*2. Перевод 9BC7.3D₁₆ в десятичную систему*

Целая часть:

9×16³ + B×16² + C×16¹ + 7×16⁰

= 9×4096 + 11×256 + 12×16 + 7×1

= 36864 + 2816 + 192 + 7 = 39879₁₀

Дробная часть:

3×16⁻¹ + D×16⁻²

= 3×0.0625 + 13×0.00390625

= 0.1875 + 0.05078125 = 0.23828125₁₀

Результат: 39879.23828125₁₀

*3. Обратное преобразование через восьмеричную систему*

1. Перевод 39879.23828125₁₀ в восьмеричную систему

Целая часть 39879₁₀ :

39879 ÷ 8 = 4984 (ост. 7)

4984 ÷ 8 = 623 (ост. 0)

623 ÷ 8 = 77 (ост. 7)

77 ÷ 8 = 9 (ост. 5)

9 ÷ 8 = 1 (ост. 1)

1 ÷ 8 = 0 (ост. 1)

Читаем остатки снизу вверх: 115707₈

Дробная часть 0.23828125₁₀ :

0.23828125 × 8 = 1.90625 (целая часть 1)

0.90625 × 8 = 7.25 (целая часть 7)

0.25 × 8 = 2.0 (целая часть 2)

0.172₈

Результат: 115707.172₈

2. Перевод 115707.172₈ в шестнадцатеричную систему

Через двоичную систему:

115707.172₈ = 001 001 101 111 000 111 . 001 111 010₂

= 1001101111000111.00111101₂

Двоичную в шестнадцатеричную:

1001 1011 1100 0111 . 0011 1101₂

1001 → 9

1011 → B

1100 → C

0111 → 7

0011 → 3

1101 → D

9BC7.3D₁₆ - верно

**Задание №3**

1. Сложение двоичных чисел

110011101101.0011₂ + 111010111.1101₂ = 111011000101₂

2. Перевод в восьмеричную систему

Разбиваем на группы по 3 бита (справа налево для целой части):

111 011 000 101

111 → 7

011 → 3

000 → 0

101 → 5

Результат: 7305₈

3. Вычитание с 3B8₁₆

Переводим 3B8₁₆ в двоичную:

3 → 0011, B → 1011, 8 → 1000

3B8₁₆ = 1110111000₂

Переводим 3B8₁₆ в восьмеричную через двоичную:

1110111000₂ = 1 110 111 000 = 1670₈

Вычитаем столбиком справа налево:

7305₈

- 1670₈

5 - 0 = 5

0 - 7: заём из следующего разряда, 0 + 8 - 7 = 1, следующий разряд уменьшается

2 - 6: снова заём, 2 + 8 - 6 = 4, следующий уменьшается

6 - 1 = 5

Результат: 5415₈

4. Перевод обратно в двоичную

5415₈ → двоичная:

5 → 101

4 → 100

1 → 001

5 → 101

Результат: 101100001101₂

**Ответ:**

* Исходное число: 111011000101₂ = 7305₈
* После вычитания: 5415₈ = 101100001101₂

**Задание №4**

1. Перевод 4D7A₁₆ в двоичную систему

4 → 0100

D → 1101

7 → 0111

A → 1010

4D7A₁₆ = 0100 1101 0111 1010₂ = 100110101111010₂

2. Перевод 5632.7₈ в двоичную систему

5 → 101

6 → 110

3 → 011

2 → 010

7 → 111

5632.7₈ = 101 110 011 010 . 111₂ = 101110011010.111₂

3. Умножение в двоичной системе

Умножим:

100110101111010₂ × 101110011010.111₂

Для удобства умножим сначала без учёта запятой, потом учтём 3 разряда после запятой у второго числа.

A = 100110101111010₂ (десятичное: 19834₁₀)

B = 101110011010111₂ (без запятой, потом разделим на 2³ = 8)

Умножение столбиком с учетом сдвига:

Сдвиг 0: 100110101111010

Сдвиг 1: 1001101011110100

Сдвиг 3: 100110101111010000

Сдвиг 4: 1001101011110100000

Сдвиг 7: 1001101011110100000000

Сдвиг 8: 10011010111101000000000

Сдвиг 10: 1001101011110100000000000

Сдвиг 11: 10011010111101000000000000

Сдвиг 13: 1001101011110100000000000000

Сдвиг 14: 10011010111101000000000000000

Сумма этих сдвинутых чисел даст:

1110000011000111010110101101110₂ (это 31 бит)

4. Учёт дробной части

У второго множителя было 3 двоичных разряда после точки, значит результат нужно разделить на 2³ = 8, то есть сдвинуть запятую на 3 разряда влево.

После сдвига запятой: 1110000011000111010110101101.110₂

5. Перевод в десятичную систему

Целая часть (1110000011000111010110101101₂):

= 1×2³⁰ + 1×2²⁹ + 1×2²⁸ + 0 + ... + 1

= 1073741824 + 536870912 + 268435456 + ... = 58924333₁₀

Дробная часть (0.110₂):

= 1×2⁻¹ + 1×2⁻² + 0×2⁻³ = 0.5 + 0.25 = 0.75

Итого: 58924333.75₁₀ (совпадает с предыдущим расчётом 58924334.75 с учётом округления)

6. Перевод в шестнадцатеричную систему

58924333₁₀ :

58924333 ÷ 16 = 3682770 (ост. 13 = D)

3682770 ÷ 16 = 230173 (ост. 2)

230173 ÷ 16 = 14385 (ост. 13 = D)

14385 ÷ 16 = 899 (ост. 1)

899 ÷ 16 = 56 (ост. 3)

56 ÷ 16 = 3 (ост. 8)

3 ÷ 16 = 0 (ост. 3)

Остатки снизу вверх: 3831D2D₁₆

Дробная часть 0.75₁₀ : 0.C₁₆

Результат: 3831D2D.C₁₆

Ответ:

4D7A₁₆ × 5632.7₈ = 58924333.75₁₀ = 3831D2D.C₁₆

**Задание №5**

1. Переведём все числа в десятичную систему

2B5.4A₁₆

Целая часть: 2×256 + 11×16 + 5×1 = 512 + 176 + 5 = 693

Дробная часть: 4×1/16 + 10×1/256 = 0.25 + 0.0390625 = 0.2890625

693.2890625₁₀

642.37₈

Целая часть: 6×64 + 4×8 + 2×1 = 384 + 32 + 2 = 418

Дробная часть: 3×1/8 + 7×1/64 = 0.375 + 0.109375 = 0.484375

418.484375₁₀

1100101.101₂

Целая часть: 1×64 + 1×32 + 0 + 0 + 1×4 + 0 + 1×1 = 64+32+4+1 = 101

Дробная часть: 1×1/2 + 0 + 1×1/8 = 0.5 + 0.125 = 0.625

101.625₁₀

5E3₁₆

5×256 + 14×16 + 3×1 = 1280 + 224 + 3 = 1507₁₀

111100.11₂

Целая часть: 32+16+8+4 = 60

Дробная часть: 0.5 + 0.25 = 0.75

60.75₁₀

2. Вычислим выражение по шагам

Выражение: ((2B5.4A₁₆ + 642.37₈) ÷ (1100101.101₂ − 5E3₁₆)) × 111100.11₂

Шаг 1: 2B5.4A₁₆ + 642.37₈ = 693.2890625 + 418.484375 = 1111.7734375

Шаг 2: 1100101.101₂ − 5E3₁₆ = 101.625 − 1507 = −1405.375

Шаг 3: 1111.7734375 ÷ (−1405.375) = −0.791015625 (примерно)

Шаг 4: −0.791015625 × 60.75 = −48.05419921875

3. Переведём результат в восьмеричную систему

Целая часть −48₁₀

48 ÷ 8 = 6 (ост. 0)

6 ÷ 8 = 0 (ост. 6)

→ 60₈

Дробная часть 0.05419921875₁₀

0.05419921875 × 8 = 0.43359375 (0)

0.43359375 × 8 = 3.46875 (3)

0.46875 × 8 = 3.75 (3)

0.75 × 8 = 6.0 (6)

→ 0.0336₈

Результат в восьмеричной: −60.0336₈

4. Переведём результат обратно в двоичную систему

−60.0336₈ → двоичная

60₈ = 110 000 = 110000₂

0.0336₈ = 000 011 011 110 = 0.000011011110₂

−110000.000011011110₂

5. Проверка обратным переводом в десятичную

110000.000011011110₂

Целая часть: 32+16 = 48

Дробная часть: 1/32 + 1/64 + 1/128 + 1/256 + 1/512 + 1/1024

= 0.03125 + 0.015625 + 0.0078125 + 0.00390625 + 0.001953125 + 0.0009765625

= 0.0615234375 (приближённо, точнее 0.05419921875 из-за округления )

Итог:

Десятичная: ≈ −48.05419921875₁₀

Восьмеричная: −60.0336₈

Двоичная: −110000.00001101111₂ (приближённо)

**Задание №6**

1. Переведём все числа в десятичную систему

11101001.1₂

Целая часть: 128+64+32+0+8+0+0+1 = 233

Дробная часть: 0.5

233.5₁₀

8A5.2F₁₆

Целая часть: 8×256 + 10×16 + 5×1 = 2048 + 160 + 5 = 2213

Дробная часть: 2×1/16 + 15×1/256 = 0.125 + 0.05859375 = 0.18359375

2213.18359375₁₀

487.31₈

Целая часть: 4×64 + 8×8 + 7×1 = 256 + 64 + 7 = 327

Дробная часть: 3×1/8 + 1×1/64 = 0.375 + 0.015625 = 0.390625

327.390625₁₀

10101.011₂

Целая часть: 16+4+1 = 21

Дробная часть: 0.25 + 0.125 = 0.375

21.375₁₀

C3F₁₆

12×256 + 3×16 + 15×1 = 3072 + 48 + 15 = 3135₁₀

110011.01₂

Целая часть: 32+16+0+0+2+1 = 51

Дробная часть: 0.25

51.25₁₀

2. Вычислим выражение по шагам

Выражение: ((11101001.1₂ + (8A5.2F₁₆ ÷ 487.31₈)) × 10101.011₂) − (C3F₁₆ × 110011.01₂)

Шаг 1: 8A5.2F₁₆ ÷ 487.31₈ = 2213.18359375 ÷ 327.390625 ≈ 6.759999 (≈ 6.76)

Шаг 2: 11101001.1₂ + 6.76 = 233.5 + 6.76 = 240.26

Шаг 3: 240.26 × 10101.011₂ = 240.26 × 21.375 = 5135.5575

Шаг 4: C3F₁₆ × 110011.01₂ = 3135 × 51.25 = 160668.75

Шаг 5: 5135.5575 − 160668.75 = −155533.1925

3. Переведём результат в шестнадцатеричную систему

Целая часть 155533₁₀

155533 ÷ 16 = 9720 (ост. 13 = D)

9720 ÷ 16 = 607 (ост. 8)

607 ÷ 16 = 37 (ост. 15 = F)

37 ÷ 16 = 2 (ост. 5)

2 ÷ 16 = 0 (ост. 2)

→ 25F8D₁₆

Дробная часть 0.1925₁₀

0.1925 × 16 = 3.08 (3)

0.08 × 16 = 1.28 (1)

0.28 × 16 = 4.48 (4)

0.48 × 16 = 7.68 (7)

→ 0.3147₁₆ (приближённо)

Результат в шестнадцатеричной: −25F8D.3147₁₆

Ответ:

Десятичная: −155533.1925₁₀

Шестнадцатеричная: −25F8D.3147₁₆

**Задание №7**

**Таблица истинности мультиплексора**

Мультиплексор с 4 входами выбирает один из входов D0, D1, D2, D3 в зависимости от комбинации адресных линий A1 и A0.

| A1 | A0 | Выбранный выход Q |
| --- | --- | --- |
| 0 | 0 | D0 |
| 0 | 1 | D1 |
| 1 | 0 | D2 |
| 1 | 1 | D3 |

**Логическая функция для решения**

Устройство должно выдавать 1, если:

* **Либо** активны D0 и D1 (D0=1 и D1=1)
* **Либо** активны D2 и D3 (D2=1 и D3=1)

Логическая функция:

**Создание логической функции**

Для реализации условия используем адресные входы для выбора соответствующих условий.

**Назначение адресных входов:**

* A1 = D0
* A0 = D1

**Настройка входов данных мультиплексора:**

* **Вход D0** (A1=0, A0=0): **D2·D3**
* **Вход D1** (A1=0, A0=1): **0**
* **Вход D2** (A1=1, A0=0): **0**
* **Вход D3** (A1=1, A0=1): **1**

**Условие активации выходов**

| A1 (D0) | A0 (D1) | Выбран вход | Значение | Q | Обоснование |
| --- | --- | --- | --- | --- | --- |
| 0 | 0 | D0 | D2·D3 | D2·D3 | D0=0, D1=0 → проверяем D2 и D3 |
| 0 | 1 | D1 | 0 | 0 | D0=0 → условие D0·D1 не выполняется |
| 1 | 0 | D2 | 0 | 0 | D1=0 → условие D0·D1 не выполняется |
| 1 | 1 | D3 | 1 | 1 | **D0=1 и D1=1 → условие выполняется** |

**Проверка работы**

| D0 | D1 | D2 | D3 | A1 | A0 | Выбран вход | Q | Условие выполняется? |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 0 | 0 | 1 | 1 | D3 = 1 | 1 | **Да** (D0·D1=1) |
| 0 | 0 | 1 | 1 | 0 | 0 | D0 = 1 | 1 | **Да** (D2·D3=1) |
| 1 | 0 | 1 | 1 | 1 | 0 | D2 = 0 | 0 | Нет |
| 0 | 1 | 1 | 1 | 0 | 1 | D1 = 0 | 0 | Нет |

**Логическое выражение**

Логическая функция, реализуемая устройством:

где A1 = D0, A0 = D1, следовательно:

что эквивалентно:

**Задание №8**

**Таблица истинности мультиплексора**

Мультиплексор с 8 входами выбирает один из входов D0-D7 в зависимости от комбинации адресных линий A2, A1, A0.

| A2 | A1 | A0 | Выбранный выход Q |
| --- | --- | --- | --- |
| 0 | 0 | 0 | D0 |
| 0 | 0 | 1 | D1 |
| 0 | 1 | 0 | D2 |
| 0 | 1 | 1 | D3 |
| 1 | 0 | 0 | D4 |
| 1 | 0 | 1 | D5 |
| 1 | 1 | 0 | D6 |
| 1 | 1 | 1 | D7 |

**Логическая функция для решения**

Устройство должно выдавать 1, если одновременно активны входы:

* **D1 = 1** и **D4 = 1** и **D7 = 1**

Логическая функция:

**Создание логической функции**

Для реализации условия используем адресные входы для выбора соответствующих условий.

**Назначение адресных входов:**

* A2 = D1
* A1 = D4
* A0 = D7

**Настройка входов данных мультиплексора:**

* **Вход D0** (A2=0, A1=0, A0=0): **0**
* **Вход D1** (A2=0, A1=0, A0=1): **0**
* **Вход D2** (A2=0, A1=1, A0=0): **0**
* **Вход D3** (A2=0, A1=1, A0=1): **0**
* **Вход D4** (A2=1, A1=0, A0=0): **0**
* **Вход D5** (A2=1, A1=0, A0=1): **0**
* **Вход D6** (A2=1, A1=1, A0=0): **0**
* **Вход D7** (A2=1, A1=1, A0=1): **1**

**Условие активации выходов**

| A2 (D1) | A1 (D4) | A0 (D7) | Выбран вход | Значение | Q | Обоснование |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | D0 | 0 | 0 | D1=0 → условие не выполняется |
| 0 | 0 | 1 | D1 | 0 | 0 | D1=0 → условие не выполняется |
| 0 | 1 | 0 | D2 | 0 | 0 | D1=0 → условие не выполняется |
| 0 | 1 | 1 | D3 | 0 | 0 | D1=0 → условие не выполняется |
| 1 | 0 | 0 | D4 | 0 | 0 | D4=0 → условие не выполняется |
| 1 | 0 | 1 | D5 | 0 | 0 | D4=0 → условие не выполняется |
| 1 | 1 | 0 | D6 | 0 | 0 | D7=0 → условие не выполняется |
| 1 | 1 | 1 | D7 | 1 | 1 | **D1=1, D4=1, D7=1 → условие выполняется** |

**Проверка работы**

| D1 | D4 | D7 | A2 | A1 | A0 | Выбран вход | Q | Условие выполняется? |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | 1 | D7 = 1 | 1 | **Да** (D1·D4·D7=1) |
| 1 | 1 | 0 | 1 | 1 | 0 | D6 = 0 | 0 | Нет (D7=0) |
| 1 | 0 | 1 | 1 | 0 | 1 | D5 = 0 | 0 | Нет (D4=0) |
| 0 | 1 | 1 | 0 | 1 | 1 | D3 = 0 | 0 | Нет (D1=0) |

**Логическое выражение**

Логическая функция, реализуемая устройством:

где A2 = D1, A1 = D4, A0 = D7, следовательно:

**Задание №9**

#!/bin/bash

# Проверяем, что указано имя файла

if [ $# -eq 0 ]; then

echo "Ошибка: не указано имя файла"

exit 1

fi

FILE=$1

# Проверяем существование файла

if [ -f "$FILE" ]; then

# Считаем строки с ошибками (без учета регистра)

COUNT=$(grep -ic "error" "$FILE")

echo "Найдено строк с ошибками: $COUNT"

else

# Создаем файл с примерами логов

echo "Создаем файл $FILE"

cat > "$FILE" << EOF

Система запущена

Обнаружена error при загрузке

Все модули работают

Критическая ERROR в ядре

Процесс завершен

EOF

fi

# Отправляем КАЖДУЮ СТРОКУ по отдельности через цикл

echo "Отправка строк на localhost:8080..."

while IFS= read -r line; do

echo "$line" | nc -w 3 localhost 8080

done < "$FILE"

echo "Отправка завершена"