



ТЕХНОЛОГИЧНО УЧИЛИЩЕ ЕЛЕКТРОННИ СИСТЕМИ
към ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ

ДИПЛОМНА РАБОТА

по професия код 523050 „Техник на компютърни системи“
специалност код 5230502 „Компютърни мрежи“

Тема: Проектиране на декодер в интегрална форма

Дипломант:

Чавдара Станиславова Белева

Дипломен ръководител:

инж. Николай Крачанов

СОФИЯ

2025

УВОД

Развитието на съвременната микроелектроника е неразривно свързано с постоянното усъвършенстване на интегралните схеми и техните градивни елементи. Декодерите, като основни комбинационни устройства, играят ключова роля в цифровата електроника, намирайки приложение в множество области - от компютърните системи до потребителската електроника.

В съвременния етап на развитие на микроелектрониката, проектирането на интегрални схеми изисква познаване на физическите процеси и явления в полупроводниковите структури, както и на съвременните методи и средства за проектиране. Особено внимание се отделя на оптимизацията на схемните решения с цел постигане на по-висока производителност, по-ниска консумация на енергия и оптимизиране на силициевата площ на интегралната схема.

Предмет на настоящата дипломна работа е проектирането на декодер 74НС42 - BCD към десетичен преобразувател (BCD-to-Decimal Decoder). Този тип декодер преобразува 4-битов BCD код в десет изходни линии, като активира съответния изход според входната комбинация. Декодерът 74НС42 намира широко приложение в управляващи системи, дисплеи и други интерфейсни схеми.

Проектирането на декодер в интегрална форма представлява комплексна задача, включваща няколко основни етапа - от първоначалното схемно решение, през симулация и верификация, до създаването на топологичен проект. Всеки от тези етапи изисква прецизен подбор на методи и средства за проектиране, отчитайки технологичните ограничения и възможности.

Съвременните системи за автоматизирано проектиране предоставят мощни инструменти за реализация на всички етапи от проектирането, позволявайки оптимизация на параметрите и характеристиките на устройството.

Цел на дипломната работа:

Основната цел на настоящата дипломна работа е проектирането на декодер 74НС42 в интегрална форма, като се използват съвременни методи и средства за проектиране, с оглед постигане на оптимални параметри на устройството.

Задачи на дипломната работа:

1. Да се проучат методите за проектиране в микроелектрониката
2. Да се поучат схемите на необходимите логически компоненти
3. Да се проектират електрическите схеми, символно изобразяване и топология на логическите компоненти
4. Да се проектира принципна електрическа схема на декодер
5. Да се проектира топология на декодер
6. Да се извършат необходимите проверки върху електрическите схеми и топологиите

Използвани съкращения

1. BCD – Binary-coded decimal (class)
2. CMOS – Complementary metal-oxide semiconductor
3. TTL – Transistor-transistor logic
4. LED – Light emitting diode
5. LCD – Liquid crystal display
6. LS – Low power Shottky (variants)
7. HC – High speed CMOS (variants)
8. CS – Chip Select
9. BI – Blanking Input
10. RBI - Ripple Blanking Input
11. RBO - Ripple Blanking Output
12. LE - Latch Enable
13. GS - Group Select
14. EO - Enable Out
15. VFD – Vacuum fluorescent display
16. PWM – Pulse-width modulation
17. FPGA – Field-programmable gate array
18. IoT – Internet of Things
19. ALU - Arithmetic Logic Unit
20. SR – Set-Reset
21. JK - Jack Kilby
22. D – Data
23. T – Toggle
24. ROM - Read-Only Memory
25. RAM - Random Access Memory
26. EEPROM - Electrically Erasable Programmable ROM
27. VOH - Output High Voltage
28. VOL - Output Low Voltage

- 29. VIH - Input High Voltage
- 30. VIL - Input Low Voltage
- 31. IP - Intellectual Property
- 32. CAD - Computer-Aided Design
- 33. VHDL - VHSIC Hardware Description Language
- 34. VHSIC - Very High-Speed Integrated Circuit
- 35. RTL - Register Transfer Level
- 36. DFT - Design for Testability
- 37. DRC - Design Rule Check
- 38. LVS - Layout Versus Schematic
- 39. PDK - Process Design Kit
- 40. ADE - Analog Design Environment
- 41. AC – Alternating current
- 42. DC – Direct current
- 43. MOSFET - Metal-oxide-semiconductor field-effect transistor
- 44. I/O – Input/Output
- 45. PVT – Process, Voltage, Temperature
- 46. LEF - Library Exchange Format
- 47. ESD – Electrostatic Discharge
- 48. CMP - Chemical Mechanical Polishing/Planarization

ПЪРВА ГЛАВА

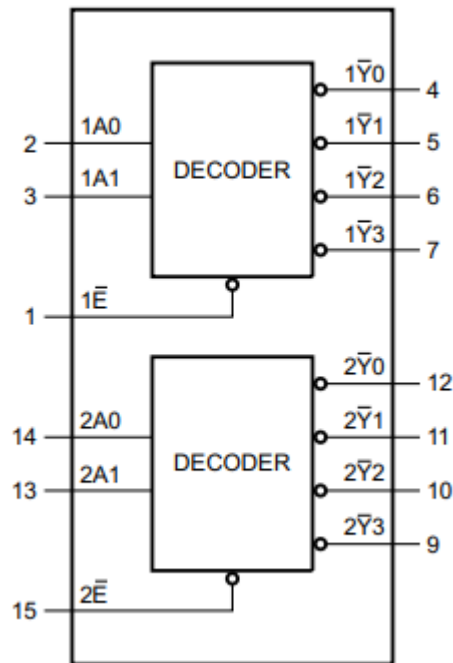
Теоретично проучване

1.1 Декодер – аналози и решения

1.1.1 Видове декодери

1.1.1.1 Two-to-four Line Decoder (Двоично-към-четворично декодиране)

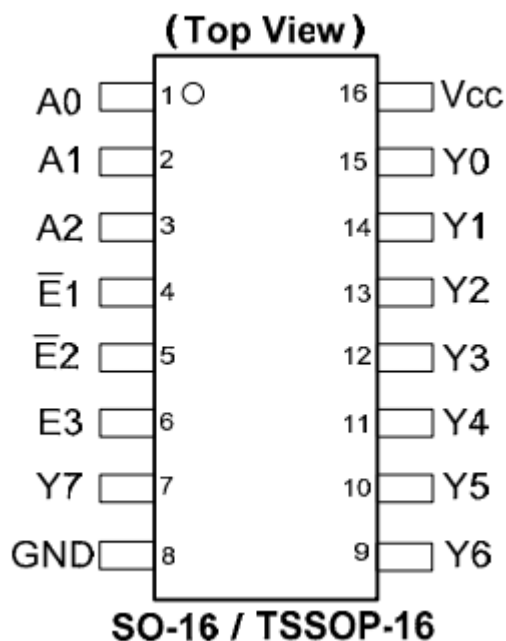
Двоично-към-четворичният декодер преобразува двубитов входен код в четири отделни изходни линии. В своята базова форма се състои от логически елементи, които преобразуват входната комбинация в активен сигнал на съответния изход. За всяка двубитова комбинация на входа (00, 01, 10 или 11) се активира точно един от четирите изхода. Схемата включва Enable вход за контрол на работата на декодера. Изходните сигнали могат да бъдат активно високи или активно ниски в зависимост от конкретната реализация. Основна интегрална схема, реализираща този тип декодер е 74НС139 (двоен 2-to-4 line декодер)(*фиг.1.1*).^{[10] [18] [19]}



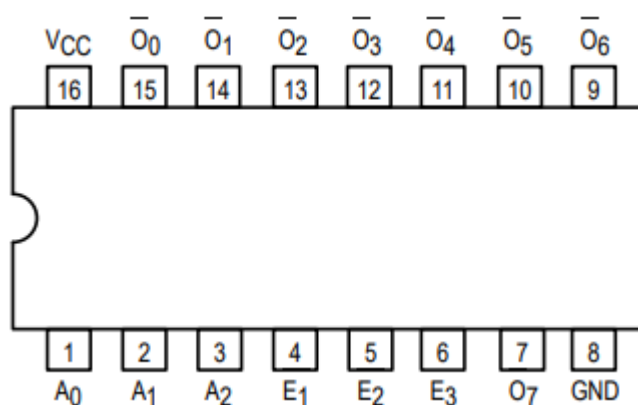
Фиг. 1.1 74HC139 (двоен 2-to-4 line декодер)

1.1.1.2 Three-to-eight Line Decoder (Троично-към-осмично декодиране)

Троично-към-осмичният декодер работи с три входни и осем изходни линии. Той приема трибитов код на входа и активира един от осемте изхода съответстващ на входната комбинация. Схемата разполага с Enable входове за контрол на работата. Този тип декодер намира приложение в системи, изискващи селекция на един от осем възможни изхода базирано на трибитов код. Най-популярните интегрални схеми са 74HC138 (CMOS версия)(*фиг. 1.2*) и 74LS138 (TTL версия)(*фиг. 1.3*), които са индустриален стандарт за този тип декодиране. ^{[3] [16] [19]}



Фиг. 1.2 74HC138 (CMOS версия)

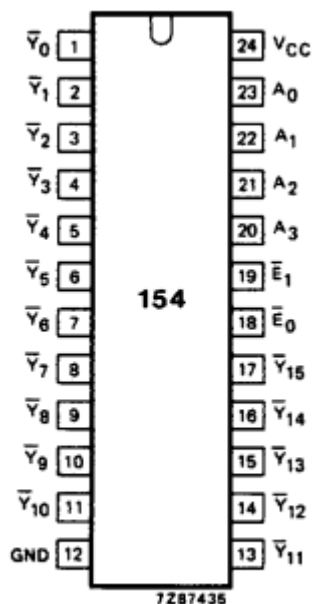


Фиг. 1.3 74LS138 (TTL версия)

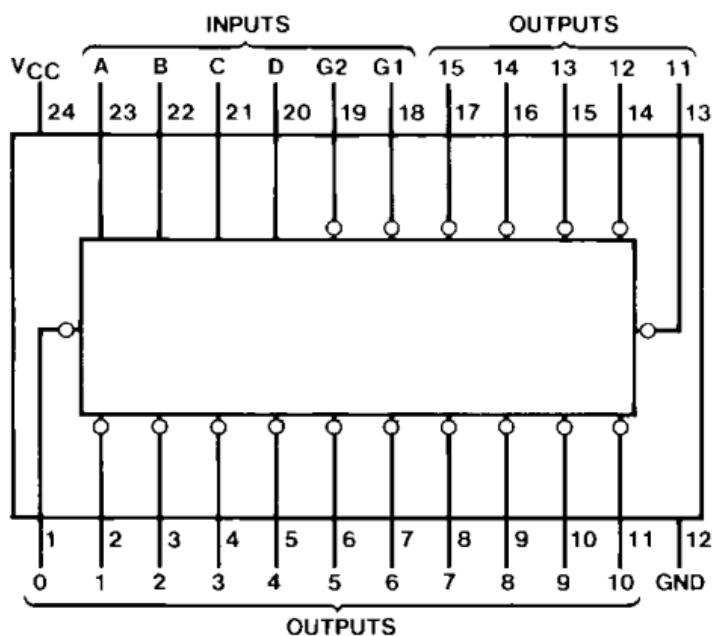
1.1.1.3 Four-to-sixteen Line Decoder (Четирибитов към 16-линиен декодер)

Четирибитовият към 16-линиен декодер приема четири входни бита и активира един от 16-те възможни изхода. Декодерът използва четирибитовата входна комбинация за избор на конкретен изход. Схемата често се реализира чрез каскадно свързване на по-прости декодери за постигане на по-голям брой изходни линии. Основните

интегрални схеми са 74HC154 (фиг. 1.4) и 74LS154 (TTL версия)(фиг. 1.5). Алтернативно, 4-to-16 декодер може да се реализира чрез каскадно свързване на два 74HC138 декодера. [4] [17] [20]



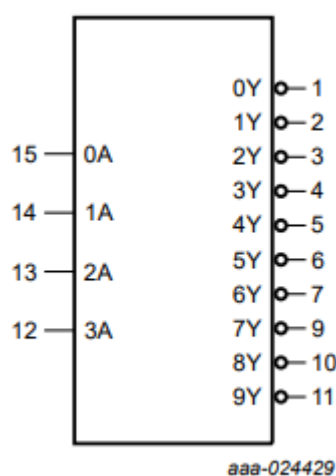
Фиг. 1.4 74HC154



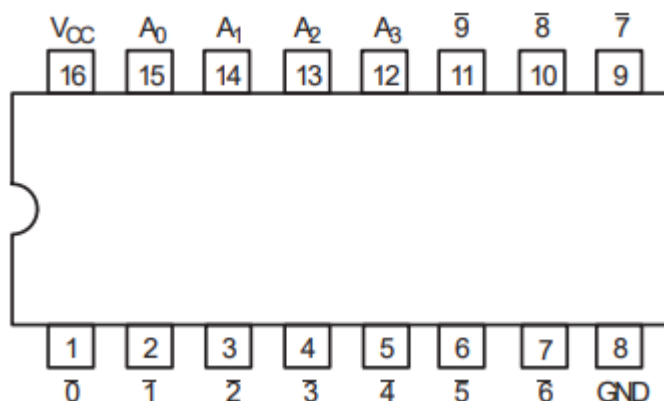
Фиг. 1.5 74LS154 (TTL версия)

1.1.1.4 BCD-to-Decimal Decoder (BCD-към-десетичен декодер)

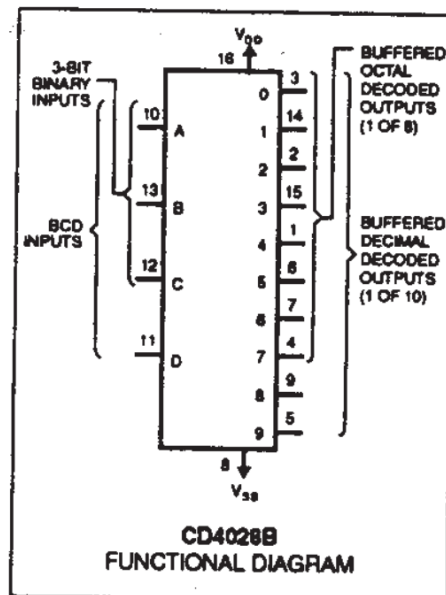
BCD-към-десетичният декодер преобразува 4-битов двоично-кодиран десетичен код в десет отделни изходни линии. Входният код обхваща стойностите от 0000 до 1001, представлящи десетичните цифри от 0 до 9. При подаване на валиден входен код се активира съответният изход. Схемата игнорира невалидните BCD кодове (1010 до 1111). Стандартните интегрални схеми включват 74HC42(фиг. 1.6), 74LS42 (TTL версия)(фиг. 1.7) и CD4028 (CMOS версия)(фиг. 1.8).^{[9] [18] [19] [21]}



Фиг. 1.6 74HC42



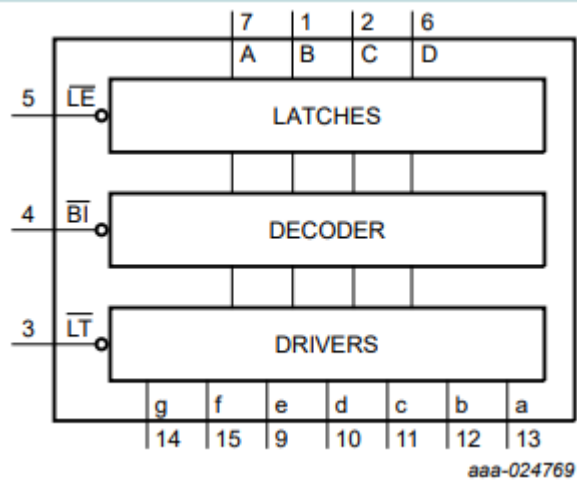
Фиг. 1.7 74LS42 (TTL версия)



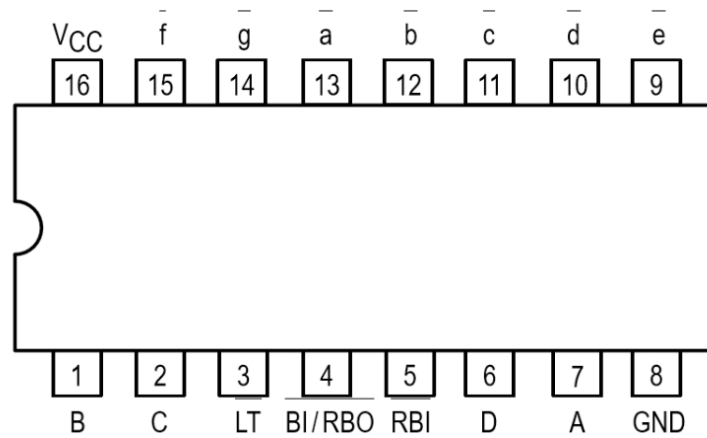
Фиг. 1.8 CD4028 (CMOS версия)

1.1.1.5 BCD-to-7-Segment Decoder (BCD-към-7-сегментен декодер)

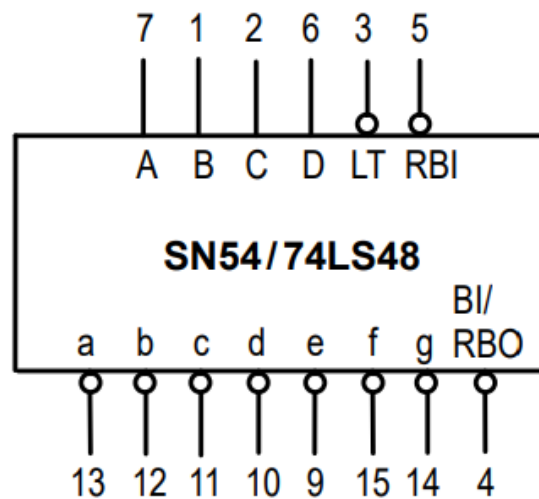
BCD-към-7-сегментният декодер преобразува 4-битов BCD код в сигнали за управление на седемте сегмента на цифров дисплей. Той генерира необходимите комбинации за визуализиране на цифри от 0 до 9. Декодерът включва функции за управление като Blanking вход и Lamp Test. Изходните сигнали се адаптират за работа с различни типове дисплеи. Основни интегрални схеми са 74HC4511 (с Latch)(*фиг. 1.9*), 74LS47 (за LED дисплеи с общ анод)(*фиг. 1.10*), 74LS48 (за LED дисплеи с общ катод)(*фиг. 1.11*) и CD4511 (CMOS версия)(*фиг. 1.12*). [3] [16] [19] [22]



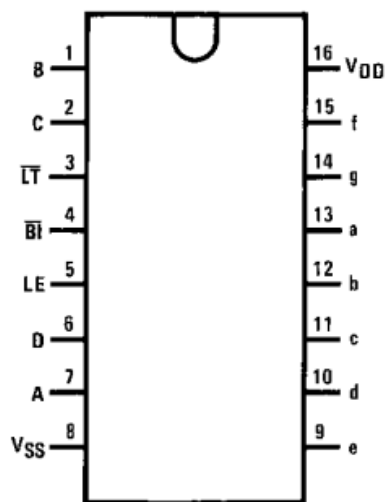
Фиг. 1.9 74HC4511 (с Latch)



Фиг. 1.10 74LS47 (за LED дисплеи с общ анод)



Фиг. 1.11 74LS48 (за LED дисплеи с общ катод)



Фиг. 1.12 CD4511 (CMOS версия)

1.1.1.6 Address Decoder (Адресен декодер)

Адресният декодер разпознава специфични адресни комбинации и генерира сигнали за избор на устройства. Той декодира част от адресната шина за генериране на сигнали за избор на чип. Тези декодери могат да работят с различни размери адресни пространства и да генерират множество селектиращи сигнали. За адресно декодиране често се използват 74НС138 за малки адресни пространства и 74НС154 за по-големи адресни пространства. В по-сложни системи могат да се използват комбинации от различни декодери за постигане на необходимата функционалност.^[1]

1.1.2 Приложения на декодерите в цифровата електроника

1.1.2.1 Приложение в компютърни системи и памети

Декодерите играят ключова роля в компютърните системи, където се използват основно за адресно декодиране. Те разпознават специфични адресни комбинации и генерират сигнали CS за

различни устройства. В системите с памет, декодерите позволяват разделянето на адресното пространство на отделни блокове, като всеки блок може да бъде селектиран независимо. Например, в 64К система памет, декодер като 74НС138 може да раздели пространството на осем 8К блока, всеки от които се активира при съответната адресна комбинация. [4] [6]

1.1.2.2 Приложение в системи за виртуализация

В областта на визуализацията, декодерите са фундаментални компоненти за преобразуване на бинарна информация във формат, подходящ за различни видове дисплеи. BCD-към-7-сегментните декодери преобразуват двоично-кодирана десетична информация в сигнали за управление на 7-сегментни LED или LCD дисплеи. При матричните дисплеи, декодерите се използват за сканиране на редове и колони, позволявайки мултиплексирано управление на големи дисплейни масиви при използване на минимален брой управляващи линии. [3] [9]

1.1.2.3 Приложение в измервателната техника

В областта на измервателната техника, декодерите са основен компонент в цифровите измервателни уреди. Те преобразуват измерените и цифровизирани стойности в подходящ формат за визуализация на дисплеи. Декодерите осигуряват преобразуването на бинарни кодове в десетични показания, което е от съществено значение за създаване на лесно четими показания на измервателните уреди. Тази функционалност е особено важна в прецизната измервателна апаратура, където точното представяне на измерените стойности е критично. [23]

1.1.2.4 Приложение в индустриални системи

В индустриалните системи декодерите намират широко приложение в контролери и автоматични машини. Те се използват за генериране на последователности от управляващи сигнали, контрол на индустриални процеси и управление на различни изпълнителни механизми. Приоритетните декодери са особено полезни в системи, където множество входни сигнали трябва да бъдат обработени според тяхната важност. В автоматизирани производствени линии, декодерите участват в управлението на различни етапи от производствения процес.^[24]

1.1.2.5 Приложение в интерфейсни системи

Декодерите са ключови компоненти в интерфейсните системи, където се изисква преобразуване между различни кодови формати. Те осигуряват съвместимост между устройства, използващи различни стандарти за представяне на данните. В системите за комуникация, декодерите могат да преобразуват данни между паралелни и последователни формати, улеснявайки обмена на информация между различни типове устройства.^[14]

1.1.2.6 Приложение в навигационни системи

В навигационните системи декодерите изпълняват няколко специфични функции. На входно ниво, те преобразуват цифровите сигнали от GPS приемници и други навигационни сензори в стандартизирани формати данни. При обработката на координати, специализираните декодери превръщат суровите данни от сателитите в географска ширина и дължина. В автомобилните

навигационни системи декодерите управляват LED или LCD дисплеи, показващи информация за скорост, посока и разстояние. Те са ключов компонент и в системите за следене на флота, където обработват данни от множество превозни средства едновременно. [4]
[23]

1.1.2.7 Приложение в роботизирани системи

В роботизираните системи декодерите се използват за управление на различни компоненти и подсистеми. Те декодират команди за управление и ги преобразуват в конкретни сигнали за управление на двигатели и други изпълнителни механизми. В по-сложни роботизирани системи, декодерите могат да участват в генерирането на сложни последователности от движения и операции, базирани на входни команди или програмирани последователности. [24]

1.1.3 Сравнение между различните видове декодери

1.1.3.1 Сравнение спрямо входно-изходното пространство

При размера на входно-изходното пространство, декодерите варират значително - от простия 74НС139 с 2 входа и 4 изхода, през средния клас 74НС138 с 3 входа и 8 изхода, до по-сложния 74НС154 с 4 входа и 16 изхода. Това създава йерархия на сложност и приложимост в различни системи. [8] [11]

1.1.3.2 Сравнение спрямо технологията на изработка

От гледна точка на технологията, съществува ясно разграничение между CMOS (НС серията) и TTL (LS серията) версиите. CMOS версиите се отличават с по-ниска консумация на енергия и по-добра

шумоустойчивост, докато TTL версиите предлагат по-високо бързодействие. Това разделение е особено важно при избора на декодер според изискванията за енергийна ефективност или бързина на работа.^{[5] [12]}

1.1.3.3 Сравнение спрямо специализацията

Съществена разлика се наблюдава и в специализацията на декодерите. Докато 74НС42 е оптимизиран за работа с BCD код и игнорира невалидни входни комбинации, 7-сегментните декодери като 74НС4511 са проектирани специално за управление на дисплеи и включват допълнителни функции като Blanking и Lamp Test. За разлика от тях, универсалните декодери като 74НС138 предлагат по-голяма гъвкавост в приложението.^[8]

1.1.3.4 Сравнение спрямо възможностите за разширяемост

Възможностите за разширяемост също варират значително. Някои декодери като 74НС138 са проектирани с възможност за лесно каскадно свързване, докато други са оптимизирани за самостоятелна работа. Това влияе съществено върху тяхната приложимост в по-големи системи.^[25]

1.1.3.5 Сравнение спрямо управляващите възможности

Управляващите възможности също се различават значително. Докато някои декодери предлагат само базови Enable входове, други включват множество контролни входове и допълнителни функции като Latch (памет) при 74НС4511. Това определя тяхната сложност на управление и възможности за интеграция в по-сложни системи.^[25]

1.1.4 Интерфейс със други цифрови компоненти

1.1.4.1 Начини на свързване с други цифрови схеми

При свързването на декодери с други цифрови схеми се използва директно свързване чрез логически нива, като често се добавят буферни елементи на входовете и изходите за подобряване на характеристиките. За защита от високи напрежения се монтират защитни диоди и ограничаващи резистори. Много декодери предлагат възможност за каскадно свързване чрез специални изводи.^[26]

1.1.4.2 Входно/изходни нива на сигналите и съвместимост

Входно-изходните нива на сигналите могат да бъдат активно високи или активно ниски според конкретното приложение. Активно ниското изходно ниво осигурява по-голяма товароспособност, което е важно при управление на LED дисплеи. Декодерите са съвместими с TTL и CMOS логически нива и имат вградена защита от пренапрежение.^[26]

1.1.4.3 Управляващи сигнали

Управлението на декодерите се осъществява чрез разнообразни контролни сигнали, които варират според типа и предназначението на декодера. Основният Enable вход активира или деактивира декодера, а CS позволява избор на конкретна интегрална схема в по-сложни системи. Някои декодери имат множество Enable входове (E1, E2, E3), което дава допълнителна гъвкавост при проектиране. При декодерите за дисплеи се използват специализирани управляващи сигнали като BI за изключване на дисплея с цел намаляване на консумацията, Lamp Test за проверка на

работоспособността на всички LED сегменти, както и RBI и RBO за управление на незначещите нули (leading zero) при каскадно свързани дисплеи. LE при декодерите с памет позволява запазване на текущите данни на изхода. Адресните декодери използват допълнителни сигнали като GS за избор на група от изходи при каскадно свързване, EO за индикация на активното състояние на декодера и Address Strobe за синхронизиране на адресната информация. Важна особеност е, че много от тези сигнали могат да бъдат активни във високо или ниско ниво (active-high или active-low), което се означава със символ над името на сигнала. [26]

1.1.4.4 Каскадно свързване на декодери

Каскадното свързване е важна характеристика на декодерите, позволяваща разширяване на функционалността чрез последователно свързване на два или повече декодера. При този метод изходите или управляващите сигнали на първия декодер се подават към входовете на следващия, което позволява увеличаване на броя изходи - например свързването на два 74HC138 декодера за получаване на 4-към-16 линеен декодер. При управление на многоразрядни дисплеи, каскадното свързване използва RBO и RBI сигналите за потискане на незначещи нули и синхронизация между отделните разряди. В системите с памет, каскадната организация позволява първият декодер да избира група от устройства, а следващият - конкретно устройство в групата, като се използват сигналите EO и GS за координация. Специфичните схеми на свързване включват свързване на Enable изхода на първия декодер към Enable входа на втория, използване на допълнителни управляващи сигнали между декодерите и възможност за добавяне на буферни елементи между тях за подобряване на характеристиките. Тази техника на свързване значително разширява

възможностите на декодерите и позволява изграждането на по-сложни системи за декодиране и управление. [26]

1.1.4.5 Интерфейс с различни видове дисплеи и индикатори

Декодерите са проектирани с гъвкав интерфейс, позволяващ им да управляват различни видове дисплеи и индикатори, като най-често се използват за директно управление на 7-сегментни LED индикатори. При тези приложения, декодерите поддържат два основни варианта на свързване - за дисплеи с общ анод или общ катод. При LED дисплей с общ анод, всички аноди на LED сегментите са свързани заедно към положителното захранване, а декодерът управлява катодите на сегментите, като използва активно ниско ниво (логическа 0) за включване на съответния сегмент. Обратно, при дисплей с общ катод, всички катода са свързани към земя, а декодерът управлява анодите с активно високо ниво (логическа 1).

Важна характеристика на декодерите са вградените драйвери, които осигуряват достатъчен ток (обикновено 10-25 mA на сегмент) за директно управление на LED сегментите. Тези драйвери включват защита от претоварване и осигуряват еднакъв интензитет на светене за всички сегменти, като елиминират нуждата от допълнителни токоограничаващи резистори.

Съвременните декодери предлагат редица специални функции за подобряване на работата с дисплеи. Функцията RB е особено полезна при многоразрядни дисплеи, тъй като позволява автоматично потискане на незначещите нули в началото на числото, което подобрява четимостта. Lamp Test функцията позволява бърза проверка на работоспособността на всички сегменти. Яркостта на дисплея може да се регулира чрез широчинно-импулсна модулация

на Blanking входа, а някои декодери поддържат и показване на шестнадесетични цифри (0-9, A-F).

За по-стабилна работа много декодери разполагат с вградена памет (latch), която позволява запазване на текущата информация на дисплея и осигурява едновременно обновяване на всички сегменти, като по този начин се избягват нежелани междинни състояния при смяна на цифрите. При работа с многоразрядни дисплеи се използва каскадно свързване на декодерите за управление на отделните разряди, като синхронизацията между тях се осъществява чрез RBI/RBO сигналите. Често се прилага и мултиплексно управление за намаляване на броя необходими изводи и общата консумация на системата.

Въпреки че са оптимизирани за LED дисплеи, съвременните декодери могат да работят и с други типове индикатори като LCD дисплеи (при осигуряване на подходящи нива на сигналите), матрични LED дисплеи, VFD и различни конфигурации от дискретни светодиоди за индикация. ^[26]

1.1.4.6 Буфериране на входове/изходи

Буферирането на входовете и изходите е реализирано чрез вградени буферни схеми, които представляват специализирани логически операции с няколко важни функции. Първо, те осигуряват усилване на сигнала, като подобряват способността за управление на товари чрез увеличаване на изходния ток без промяна на логическото ниво. Второ, създават електрическа изолация между входната и изходната верига, предпазвайки декодера от смущения и обратни въздействия. Трето, спомагат за възстановяване на формата на цифровите сигнали чрез подобряване на времената за нарастване и спадане.

Буферите могат да бъдат неинвертиращи (запазват логическото ниво), инвертиращи (обръщат нивото) или три-състоянни (с

допълнително състояние "висок импеданс"). Те най-често са вградени в самия декодер, но при необходимост могат да се добавят и външни буферни елементи за допълнително подобряване на характеристиките.^[26]

1.1.4.7 Защита от шум и смущения

Декодерите са чувствителни електронни устройства, които трябва да разпознават и обработват сигнали с ниско ниво, което ги прави податливи на електрически смущения и шум от външни източници. Ако тези смущения достигнат чувствителните входове на декодера, може да се получат грешки в декодирания изходен сигнал или дори повреда на устройството. За да се избегне това, производителите вграждат специални защитни елементи в декодерите.

Първата линия на защита са така наречените защитни или клампиращи диоди. Те се свързват между всеки вход/изход на декодера и захранващите линии. При поява на пренапрежение, тези диоди ограничават напрежението до безопасни стойности, като го отвеждат към захранването. По този начин те защитават чувствителните входове от потенциално опасни високоволтови импулси.

Освен защитните диоди, в декодерите се използват и ограничаващи или балластни резистори. Те се включват последователно във всеки вход и изход и имат за цел да ограничат максималния ток, който може да протече през изводите при евентуално късо съединение. Така се избягва прегряване и повреда на чиповете. Стойностите на тези резистори се подбират внимателно, така че да не влошават качеството на преминаващия сигнал.

Като допълнителна мярка, в самите чипове на декодерите често се вграждат и вътрешни RC филтри. Тяхната роля е да потискат

високочестотните смущения и импулсен шум, които успеят да преминат през външните защитни елементи.

Въпреки вградените защиты, за постигане на максимална шумоустойчивост се препоръчва използването и на някои външни мерки. Една от тях е качествено филтриране на входните сигнали. Това може да се постигне чрез включване на нискочестотни RC или LC филтри, настроени според работната честота на декодера. Те ефективно елиминират нежеланите високочестотни смущения и изглаждат формата на сигнала.

Друг важен фактор е осигуряването на стабилно и добре филтрирано захранване за декодера. Използването на нискоимпедансни кондензатори с достатъчен капацитет спомага за изглаждане на пулсациите в захранващото напрежение. При по-високи изисквания могат да се добавят и допълнителни LC филтри или линейни стабилизатори, които ефективно потискат шума в захранващите линии.

В някои случаи се прибегва и до екраниране на особено чувствителните области с метални кутии или екрани. Това предотвратява директното индуциране на електромагнитни смущения в схемата чрез ефекта на Фарадеевия кафез.

Накрая, използването на усукани двойки проводници или екранирани кабели за входните и изходни връзки също допринася за шумоустойчивостта. Усукването потиска електромагнитното излъчване и капацитивното съгласуване между съседни проводници, а екранирането ги изолира допълнително от външни полета.

Чрез умелото комбиниране на всички тези вътрешни и външни защиты и филтриращи елементи, съвременните декодери могат да работят надеждно дори в среда с повишен електромагнитен шум и смущения. Това е от критично значение за правилното разпознаване и декодиране на слабите входни сигнали и получаването на качествен изходен резултат. ^[26]

1.1.5 Анализ на съществуващи решения и архитектури:

1.1.5.1 Преглед на популярни интегрални схеми с декодери (74xx серия)

1.1.5.1.1 74НС138/74НСТ138

74НС138/74НСТ138 е един от най-популярните и универсални 3-към-8 линейни декодери. Той разполага с 3 входа за адресиране, които управляват 8 изхода, както и с 3 допълнителни входа Enable, позволяващи каскадно свързване на няколко устройства. Изходите са активно ниски, което ги прави особено подходящи за управление на CS сигнали в цифровите системи. Типичното му приложение е в адресни декодери за памети и периферия, но може да се използва и за генериране на различни последователности.^[27]

1.1.5.1.2 74НС42/74НСТ42

74НС42/74НСТ42 е специализиран BCD към десетичен декодер за преобразуване на двоично-десетичен код в десетичен формат. Схемата приема 4-битов BCD вход в диапазона от 0000 до 1001 и активира един от десетте си активно ниски изхода, съответстващ на входната цифра. Невалидните BCD комбинации автоматично се блокират. Исторически този декодер е бил използван за управление на газоразрядни индикатори, но намира приложение и при други видове десетични дисплеи.^[27]

1.1.5.1.3 74НС154/74НСТ154

74НС154/74НСТ154 е разширен вариант на 74НС138, представляващ 4-към-16 линеен декодер. Той има 4 входа за адресиране, които управляват 16 активно ниски изхода, както

и 2 входа за разрешаване на работата. Този декодер е особено полезен при работа с големи адресни пространства, тъй като може да управлява до 16 отделни устройства. Той често се използва в системи, изискващи декодиране на 4-битови данни или адреси.^[27]

1.1.5.2 Сравнение на различни архитектури

Различните архитектури при декодерите предлагат различни компромиси между бързодействие, консумация на енергия и сложност на реализацията.

Най-простата архитектура е комбинационната логика, при която входният код директно се преобразува в изходен сигнал чрез логически елементи. Този подход предлага максимално бързодействие, тъй като закъснението се определя само от времето за разпространение на сигнала през логическите елементи. Същевременно консумацията на енергия е относително висока поради постоянната активност на всички елементи, а сложността нараства експоненциално с увеличаване на броя входове.

Каскадната архитектура представлява по-ефективен подход от гледна точка на хардуерната реализация. При нея декодирането се извършва на няколко последователни етапа, като всеки етап обработва част от входния код. Това позволява значително намаляване на броя необходими логически елементи, но за сметка на по-голямо закъснение поради последователната обработка. Консумацията на енергия е по-ниска в сравнение с комбинационната логика.

Матричната архитектура използва програмируема матрица от връзки за реализиране на декодиращата функция. Този подход предлага максимална гъвкавост и възможност за препрограмиране, но има по-ниско бързодействие поради допълнителните комутационни

елементи. Консумацията зависи силно от конкретната реализация, но обикновено е по-висока от каскадната архитектура поради наличието на програмируеми елементи.

Съвременните декодери често комбинират различни архитектурни подходи за постигане на оптимален баланс между параметрите. ^[27]

1.1.5.3 Анализ на предимства и недостатъци на всяко решение

Простите логически декодери, като 2-към-4 и 3-към-8 линейните декодери, се характеризират с минимална сложност и ниска консумация на енергия. Те са изградени от малък брой логически елементи, което ги прави евтини за производство и лесни за интегриране. Основният им недостатък е ограниченият брой изходи и липсата на допълнителна функционалност като буфериране или защита от смущения. Въпреки това, те намират широко приложение в прости системи за адресиране и управление.

BCD-към-7-сегментните декодери представляват по-сложно решение с добавена специализирана функционалност. Те включват допълнителни схеми за управление на дисплея като избор на шрифт, автоматично потискане на водещите нули и регулиране на яркостта. Това увеличава консумацията и цената, но значително улеснява изграждането на цифрови индикатори. Модерните версии с Latch допълнително повишават гъвкавостта, макар и с цената на по-висока сложност.

Приоритетните декодери добавят логика за арбитраж, която им позволява да обработват множество едновременни заявки. Това е особено полезно в системи за прекъсвания и управление на достъпа, но усложнява схемата и забавя бързодействието поради допълнителните нива на логика. Консумацията също нараства заради по-сложната обработка.

Декодерите с тристабилни изходи предлагат възможност за свързване към обща шина, което е важно в компютърните системи. Цената за това е необходимостта от допълнителни буфери и схеми за управление на разрешаването. Тези декодери са по-бавни от простите комбинационни схеми поради превключването между състоянията, но предлагат по-добра системна интеграция.

Каскадните декодери позволяват изграждането на по-големи декодиращи системи чрез свързване на няколко по-прости декодера. Това е елегантно решение за разширяване на възможностите, но води до увеличаване на закъснението на сигналите с всяко допълнително ниво. Консумацията нараства пропорционално на броя използвани декодери.^[28]

1.1.5.4 Методи за оптимизация

В съвременната цифрова електроника съществуват множество методи за оптимизация на декодерите, които позволяват подобряване на техните характеристики и ефективност.

Първият и най-важен метод за оптимизация е минимизацията на логическите изрази, описващи работата на декодера. Чрез използване на карти на Карно или други методи за минимизация може значително да се намали броят на логическите елементи в схемата. Това води не само до по-малка площ на интегралната схема, но и до намаляване на закъснението на сигналите и консумацията на енергия. Например, при 4-битов декодер често могат да се премахнат до 30% от логическите елементи без промяна във функционалността.

Друг ефективен метод е реструктурирането на декодиращата логика в дървовидна структура. Вместо директно декодиране на всички входни комбинации, входният код се разделя на групи и декодирането се извършва на няколко по-малки стъпки. Този

подход, известен като йерархично декодиране, може значително да намали общото закъснение на сигналите, особено при декодери с голям брой изходи. Допълнително предимство е възможността за изключване на неизползваните клонове на дървото, което намалява динамичната консумация.

Важен аспект на оптимизацията е използването на предварително декодиране. При този метод част от входния код се декодира предварително, докато останалата част от входните сигнали все още се установява. Това създава паралелизъм в обработката и намалява общото време за декодиране. Особено ефективен е този подход при адресни декодери в памети, където времето за достъп е критично.

Съвременните технологии позволяват и прилагането на динамично преконфигуриране на декодиращата логика. При този подход декодерът може да променя своята структура в зависимост от текущите изисквания на системата. Например, при ниско натоварване може да се активира режим с понижена консумация, а при необходимост от висока производителност - да се превключи към по-бърза конфигурация с по-висока консумация.

Не на последно място стои оптимизацията на физическото разположение на елементите в интегралната схема. Чрез внимателно планиране на топологията могат да се минимизират паразитните капацитети и да се подобрят времевите характеристики. Особено важно е това при високоскоростни декодери, където всяко допълнително закъснение е критично.^[27]

1.1.5.5 Съвременни тенденции в дизайна на декодери

Съвременният дизайн на декодери преминава през значителна трансформация, движена от нарастващите изисквания на цифровите системи и новите технологични възможности. Основна тенденция е интегрирането на все повече интелигентни функции в декодерните

схеми. Традиционното просто преобразуване на кодове се допълва с вградени буфери за данни, схеми за откриване и корекция на грешки, както и програмируеми интерфейси за гъвкава настройка на работните параметри.

В областта на дисплейните декодери наблюдаваме преход към по-комплексни решения, които поддържат множество типове индикатори и различни режими на визуализация. Съвременните BCD декодери често включват PWM управление на яркостта, автоматично мултиплексиране на няколко разряда и вградена поддръжка както за LED, така и за LCD дисплеи. Някои модели предлагат и допълнителни възможности като анимации, специални символи и диагностични функции.

Развитието на високоскоростните интерфейси налага нов подход в проектирането на декодерите за адресиране на памет и периферия. Вместо конвенционалните комбинационни схеми, все по-често се използват pipeline архитектури с предварително декодиране и кеширане на резултатите. Това позволява постигането на по-високи работни честоти при запазване на ниска консумация на енергия.

Важна тенденция е интегрирането на декодерите с други функционални блокове в рамките на системата. Модерните процесори и FPGA често включват конфигурируеми декодерни блокове, които могат да се адаптират според конкретното приложение. Това намалява необходимостта от външни компоненти и подобрява общата производителност на системата.

В областта на индустриалната автоматизация и IoT устройствата се налагат интелигентни декодери с вградени защиты и диагностични възможности. Те включват схеми за филтриране на шумове, защита от пренапрежение и претоварване, както и възможности за отдалечен мониторинг и конфигуриране. Някои модели поддържат и криптиране на данните за повишена сигурност.

Все по-голямо значение придобива и енергийната ефективност. Съвременните декодери често включват схеми за автоматично превключване в режим на ниска консумация при липса на активност, както и възможности за селективно активиране само на необходимите функционални блокове. Това е особено важно за батерийно захранваните устройства и системите за обработка на големи обеми данни.^[27]

1.1.5.6 Критерии за избор на конкретно решение според приложението

При избора на декодиращо решение е необходимо да се направи задълбочен анализ на изискванията на конкретното приложение и да се балансират множество фактори. На първо място трябва да се определи необходимата скорост на работа на декодера. В системи с високи изисквания към бързодействието, като например компютърни шини или високоскоростни комуникации, се препоръчва използването на паралелни или предварително декодиращи схеми, въпреки тяхната по-висока сложност и консумация.

Друг ключов фактор е необходимият брой входи и изходи на декодера. За приложения с малък брой сигнали, простите комбинационни декодери са напълно достатъчни. При по-големи системи може да се наложи използването на каскадни решения или по-сложни интегрални схеми. Важно е да се прецени и възможността за бъдещо разширение на системата, тъй като това може да повлияе на избора между фиксирана или модулна архитектура.

Консумацията на енергия е особено важна при преносими и захранвани от батерии устройства. В такива случаи може да се предпочетат по-прости декодери с по-ниска консумация, дори това

да означава известен компромис с бързодействието. Модерните CMOS технологии предлагат добър баланс между консумация и производителност, особено при по-ниски работни честоти.

Надеждността на декодера трябва да съответства на изискванията за безопасност и стабилност на системата. В индустриални и медицински приложения често се налага използването на декодери с вградени схеми за откриване и корекция на грешки, както и защита от електромагнитни смущения. Това увеличава сложността и цената, но е задължително за критични приложения.

Икономическите съображения също играят важна роля. Освен директната цена на компонентите, трябва да се отчетат разходите за разработка, тестване и поддръжка. Използването на стандартни декодерни схеми може да намали значително времето за разработка и да улесни отстраняването на евентуални проблеми, въпреки че понякога специализираното решение може да е по-ефективно в дългосрочен план. [27]

1.2. Цифрови логически компоненти

1.2.1 Основни логически елементи (AND, OR, NOT, NAND, NOR, XOR)

1.2.1.1 AND

AND (И) представлява логическа схема, която реализира операцията логическо умножение. При него изходният сигнал достига високо ниво само когато всички входни сигнали са във високо ниво. Това прави AND (*фиг. 1.13*)(*фиг. 1.14*) елемента особено полезен в ситуации, където е необходимо да се проверява едновременното изпълнение на множество условия. Например, в системите за сигурност, където достъпът се разрешава само при наличието на няколко валидирани входни параметъра едновременно. [3] [18]



Фиг. 1.13 Символно представяне на AND

Truth Table

A (Input 1)	B (Input 2)	X = (A.B)
0	0	0
0	1	0
1	0	0
1	1	1

Фиг. 1.14 Таблица на истинност на AND

1.2.1.2 OR

OR (ИЛИ)(фиг. 1.15)(фиг. 1.16) преминава във високо ниво, когато поне един от входовете е във високо ниво. Това свойство го прави идеален за приложения, където трябва да се открие наличието на поне едно събитие от група възможни събития. Типичен пример е системата за аларми, където задействането на който и да е сензор трябва да генерира сигнал за тревога. [7] [10] [16]



Фиг. 1.15 Символно представяне на OR

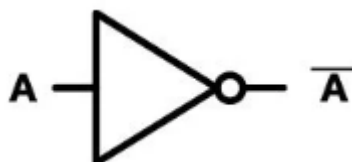
Truth Table

Input A	Input B	Output
0	0	0
0	1	1
1	0	1
1	1	1

Фиг. 1.16 Таблица на истинност на OR

1.2.1.3 NOT

NOT (НЕ) е най-простият от всички, но неговата роля е изключително важна. Той инвертира входния сигнал, превръщайки високото ниво в ниско и обратно. Това позволява създаването на комплементарни сигнали, които са необходими в множество цифрови схеми. NOT (фиг. 1.17)(фиг. 1.18) често се използва в комбинация с други логически елементи за създаване на по-сложни функции. ^{[9] [17]}



Фиг. 1.17 Символно представяне на NOT

Truth Table

A (Input)	$Y = \bar{A}$ (Output)
0	1
1	0

Фиг. 1.18 Таблица на истинност на NOT

1.2.1.4 NAND

NAND (И-НЕ) комбинира функциите на AND и NOT в един компонент. Това е един от най-важните елементи в цифровата електроника, тъй като е универсален - с помощта само на NAND (фиг. 1.19)(фиг. 1.20) елементи може да се реализира всяка друга логическа функция. Освен това, NAND елементите са по-евтини за производство от обикновените AND елементи, което ги прави предпочитан избор в интегралните схеми. ^{[7] [18]}



Фиг. 1.19 Символно представяне на NAND

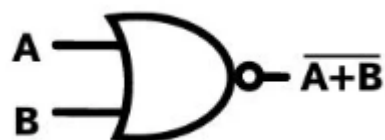
Truth Table

Input A	Input B	Output
0	0	0
0	1	1
1	0	1
1	1	1

Фиг. 1.20 Таблица на истинност на NAND

1.2.1.5 NOR

NOR (ИЛИ-НЕ) също е универсален и комбинира OR и NOT функциите. Неговият изход е във високо ниво само когато всички входове са в ниско ниво. NOR (фиг. 1.21)(фиг. 1.22) елементите са особено популярни в CMOS технологията поради тяхната ефективна реализация на транзисторно ниво. ^[11]



Фиг. 1.21 Символно представяне на NOR

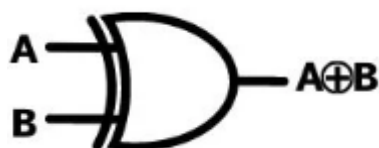
Truth Table

Input A	Input B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

Фиг. 1.22 Таблица на истинност на NOR

1.2.1.6 XOR

XOR (Изключващо ИЛИ) (фиг. 1.23)(фиг. 1.24) има специално място в цифровата електроника. Неговият изход е във високо ниво само когато входовете са в различни състояния. Това свойство го прави незаменим в аритметичните схеми, особено в суматорите, както и в схемите за проверка на четност и в криптографските хардуерни модули.^[17]



Фиг. 1.23 Символно представяне на XOR

Truth Table

A (Input 1)	B (Input 2)	$X = A'B + AB'$
0	0	0
0	1	1
1	0	1
1	1	0

Фиг. 1.24 Таблица на истинност на XOR

1.2.1.7 XNOR

XNOR (Изключващо ИЛИ-НЕ) е логическото допълнение на XOR (фиг. 1.25)(фиг. 1.26) елемента. Той се нарича още съвпадение или еквивалентност, тъй като неговият изход е във високо ниво само когато входовете са в еднакви състояния (или двата са във високо ниво, или двата са в ниско ниво). Тази характеристика прави XNOR особено полезен в схеми за сравнение и проверка на равенство между цифрови сигнали. ^[6]



Фиг. 1.25 Символно представяне на XNOR

Truth Table

Input A	Input B	Output
0	0	1
0	1	0
1	0	0
1	1	1

Фиг. 1.26 Таблица на истинност на XNOR

1.2.2 Комбинационни схеми

Комбинационните схеми представляват следващото ниво на сложност в цифровата електроника. Те се характеризират с това, че техните изходи зависят единствено от моментните стойности на входовете, без да се запаметява каквато и да било информация за предишни състояния. Това ги прави предвидими и лесни за анализ, тъй като за всяка комбинация от входни сигнали съществува точно определена комбинация от изходни сигнали.

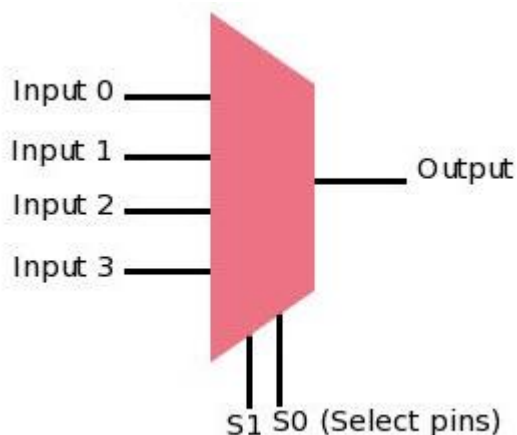
1.2.2.1 Логически примитиви

Логическите примитиви са най-простите комбинационни схеми. Те представляват реализация на основните булеви функции в електронен вид. Чрез комбиниране на различни логически гейтове могат да се създадат по-сложни функции, необходими за реализацията на по-комплексни цифрови системи. Всеки примитив се характеризира със специфични времена на превключване и закъснения, които трябва да се вземат предвид при проектирането на схемите. [3] [10]

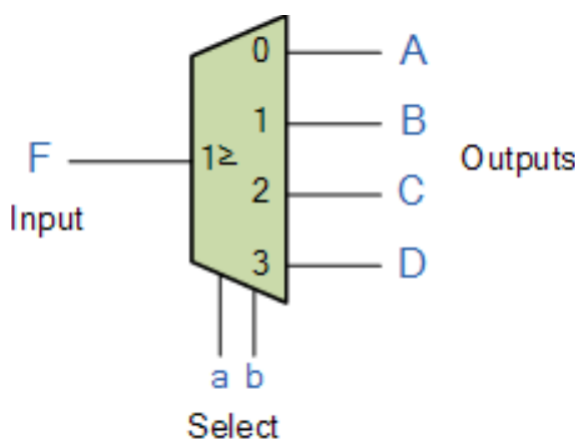
1.2.2.2 Мултиплексори и демултиплексори

Мултиплексорите (*фиг. 1.27*) и демултиплексорите (*фиг. 1.28*) са по-сложни комбинационни схеми, които играят ключова роля в предаването на данни. Мултиплексорът избира един от множество входни сигнали и го насочва към единствения си изход, базирайки се на състоянието на управляващите (адресни) входове. Това го прави изключително полезен в комуникационните системи, където множество източници на данни трябва да споделят един канал за предаване. Демултиплексорът извършва обратната операция -

разпределя един входен сигнал към един от множество изходи, отново според състоянието на адресните входове. Тези компоненти са особено важни в системите за разпределение на данни и в схемите за адресиране на памет. [4]



Фиг. 1.27 Логическа схема на мултиплексор

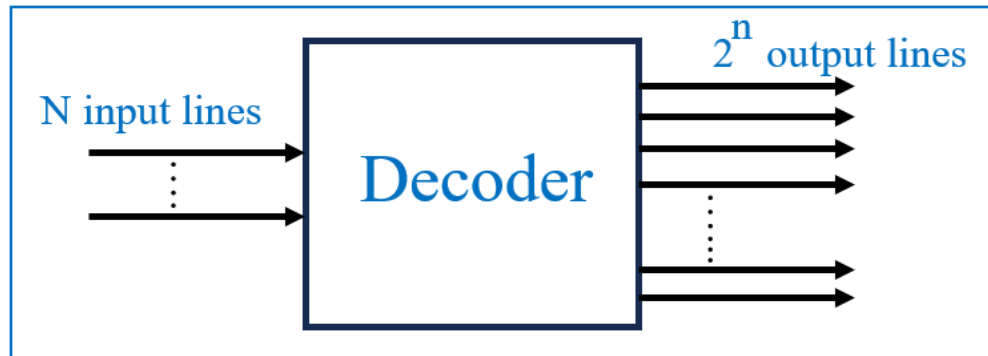


Фиг. 1.28 Логическа схема на демултиплексор

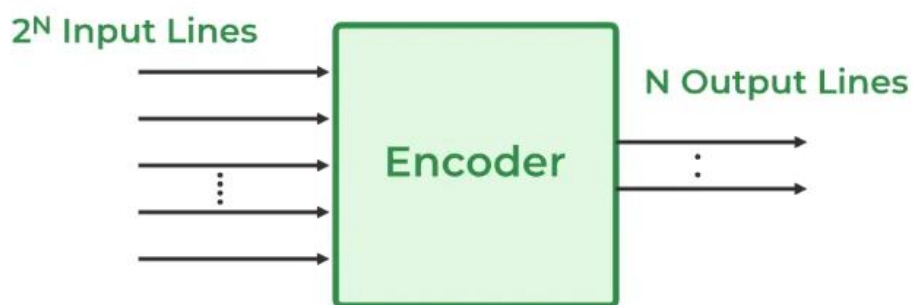
1.2.2.3 Декодери и енкодери

Декодерите (фиг. 1.29) и енкодерите (фиг. 1.30) са специализирани комбинационни схеми, които преобразуват информацията от един формат в друг. Декодерът преобразува двоичен код в един активен изход от множество възможни изходи. Класически пример е декодерът за 7-сегментен дисплей, който превръща двоичния код на

цифрата в съответната комбинация от сегменти. Енкодерът извършва обратната операция - при него един активен вход се преобразува в съответния двоичен код. Тези компоненти са незаменими в интерфейсите схеми и системите за въвеждане на данни. [7] [9] [16]



Фиг. 1.29 Логическа схема на декодер

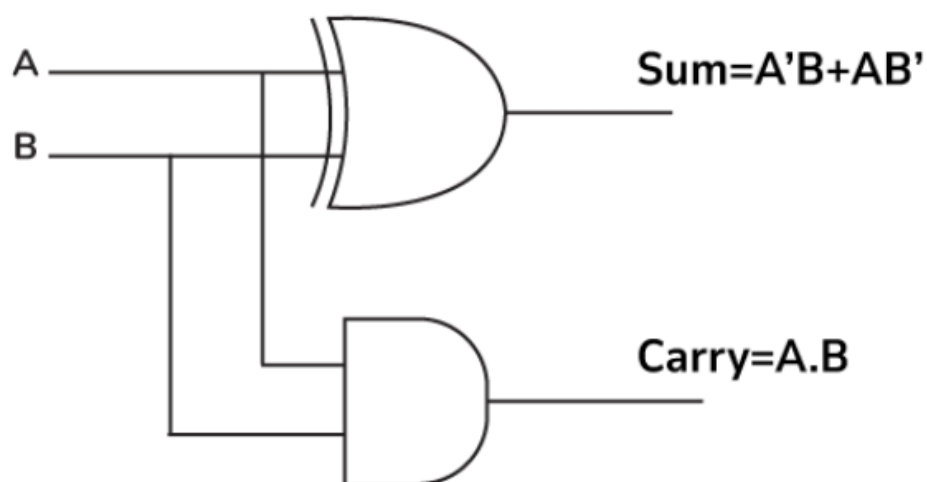


Фиг. 1.30 Логическа схема на енкодер

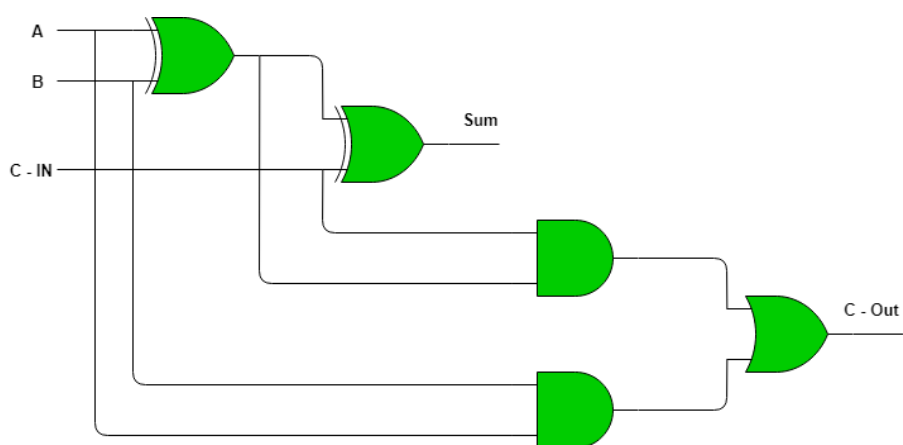
1.2.2.4 Суматори

Суматорите са специализирани комбинационни схеми, които извършват аритметично събиране на двоични числа. Те се делят на полусуматори (фиг. 1.31), които могат да събират две еднобитови числа, и пълни суматори (фиг. 1.32), които могат да обработват и входящ пренос от предишна степен. Суматорите са основни

компоненти в ALU на процесорите и се използват във всички изчислителни операции. ^{[1] [11]}



Фиг. 1.31 Логическа схема на полусуматор



Фиг. 1.32 Логическа схема на пълен суматор

1.2.2.5 Компаратори

Компараторите (фиг. 1.33) са схеми, които сравняват двоични числа и генерират изходни сигнали, показващи резултата от сравнението (равно, по-голямо или по-малко). Те са особено важни в системите

за контрол и вземане на решения, където често се налага сравняване на различни цифрови стойности. [6]



Фиг. 1.32 Логическа схема на компаратор

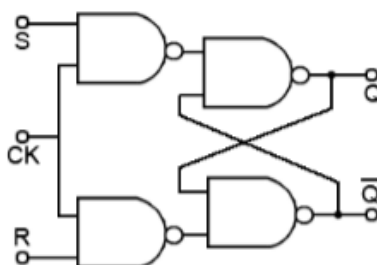
1.2.3 Последователни схеми

Последователностните схеми представляват по-високо ниво на сложност в цифровата електроника, тъй като при тях изходните сигнали зависят не само от текущите входни сигнали, но и от предишните състояния на схемата. Това свойство им позволява да "запомнят" информация и да я използват при определяне на следващите си състояния.

1.2.3.1 Тригери

Най-простият и фундаментален елемент на последователностните схеми е тригерът (flip-flop). Съществуват няколко основни типа тригери, всеки със свои специфични характеристики и приложения. SR тригерът (фиг. 1.33) е най-простият тип, имащ два входа - за установяване и нулиране. Въпреки простотата си, той има едно съществено ограничение - забранено състояние, което възниква при едновременно активиране на двата входа.

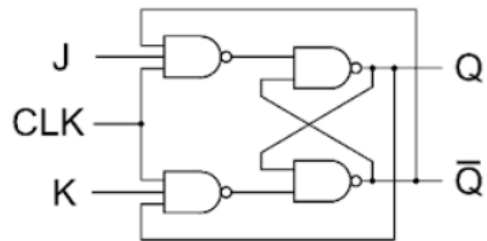
JK тригерът (фиг. 1.34) преодолява този недостатък, като елиминира забраненото състояние чрез по-сложна вътрешна логика. Той може да работи като брояч и е изключително гъвкав в приложенията си. D тригерът (фиг. 1.35) е може би най-широко използваният тип, тъй като има само един информационен вход и работи по много интуитивен начин - при всеки тактов импулс запомня стойността от D входа. T тригерът (фиг. 1.36) е специализиран за приложения, изискващи превключване на състоянието, като например броячи и схеми за деление на честота. [4]



TRUTH TABLE

S	R	Q_N	Q_{N+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	-
1	1	1	-

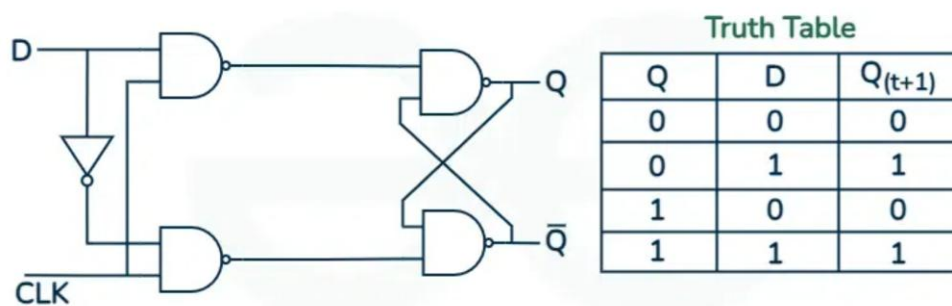
Фиг. 1.33 Логическа схема на SR flip-flop и таблица на истинност



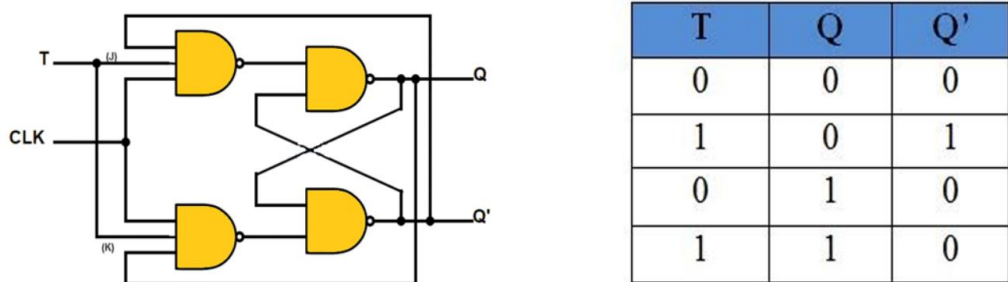
TRUTH TABLE

J	K	Q_N	Q_{N+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Фиг. 1.34 Логическа схема на JK flip-flop и таблица на истинност



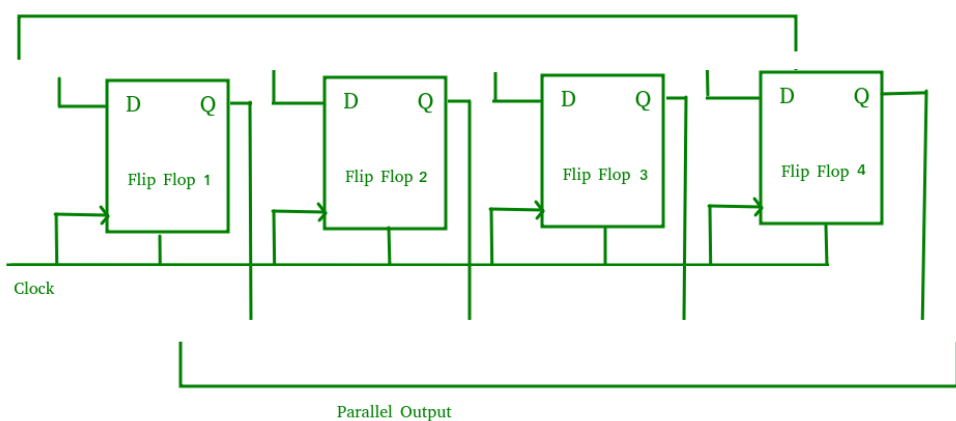
Фиг. 1.35 Логическа схема на D flip-flop и таблица на истинност



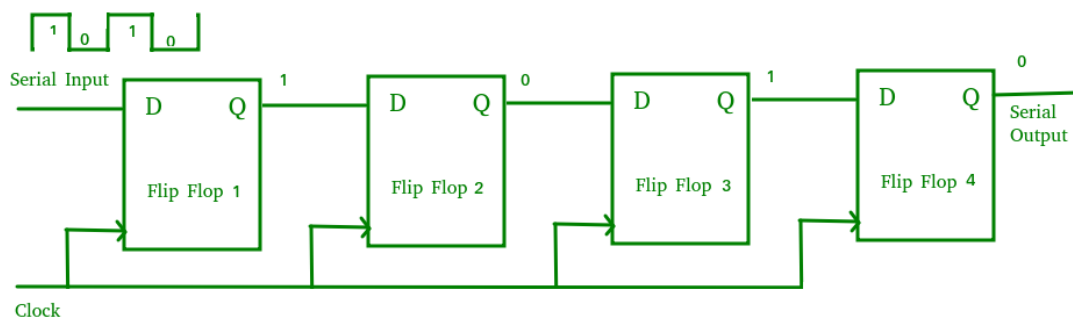
Фиг. 1.36 Логическа схема на T flip-flop и таблица на истинност

1.2.3.2 Регистри

Регистрите са по-сложни последователни схеми, изградени от група тригери, работещи заедно за съхранение на многобитови данни. Те могат да бъдат паралелни (фиг. 1.37), при които всички битове се записват едновременно, или последователни (фиг. 1.38), при които данните се въвеждат бит по бит. Универсалните регистри комбинират и двата режима на работа. Регистрите са ключови компоненти в процесорните системи, където се използват за временно съхранение на данни и междинни резултати. [3]



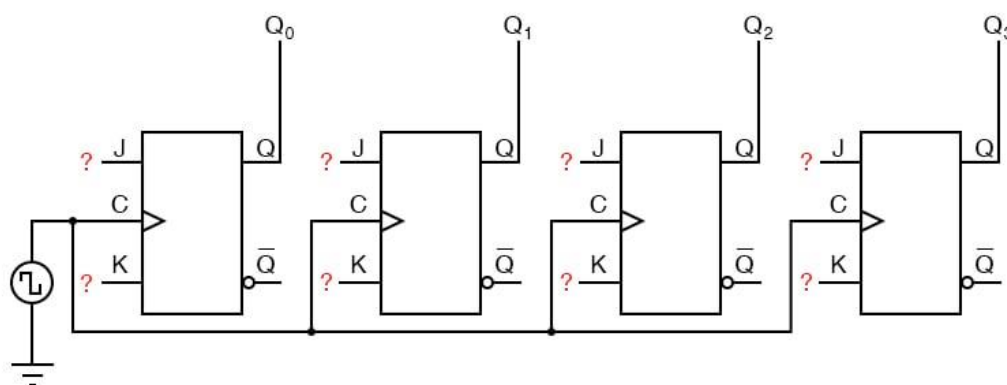
Фиг. 1.37 Логическа схема на паралелен регистър



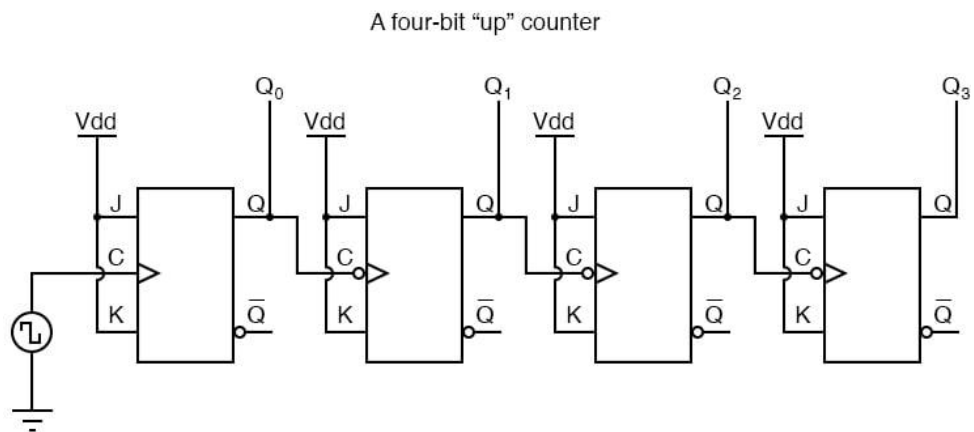
Фиг. 1.38 Логическа схема на последователен регистър

1.2.3.3 Броячи

Броячите са специализирани последователностни схеми, които изпълняват функцията на броене на импулси. Те могат да бъдат синхронни (фиг. 1.39), при които всички тригери превключват едновременно, или асинхронни (фиг. 1.40), при които превключването се разпространява последователно през веригата. Броячите са особено важни в таймери, генератори на последователности и схеми за деление на честота.^[10]



Фиг. 1.39 Логическа схема на синхронен брояч

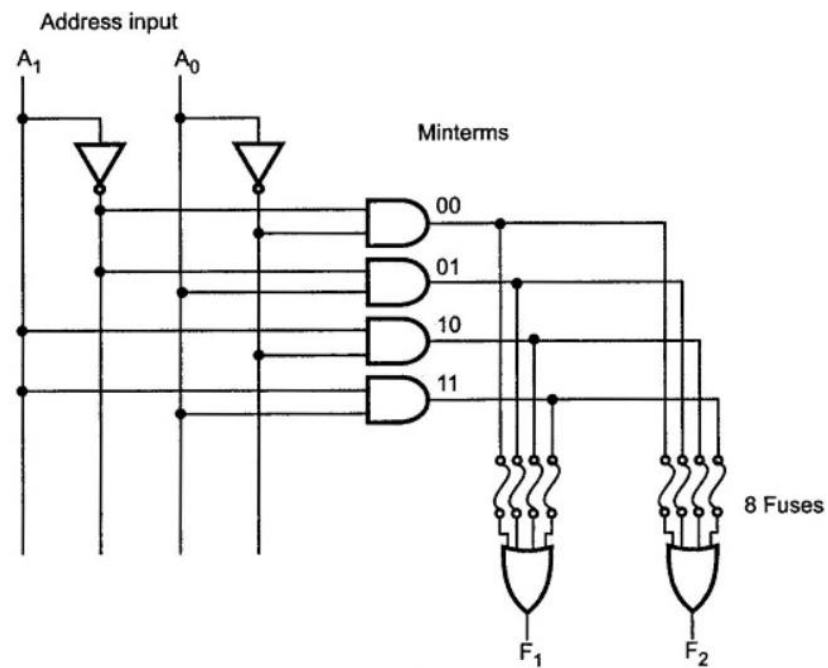


Фиг. 1.40 Логическа схема на асинхронен брояч

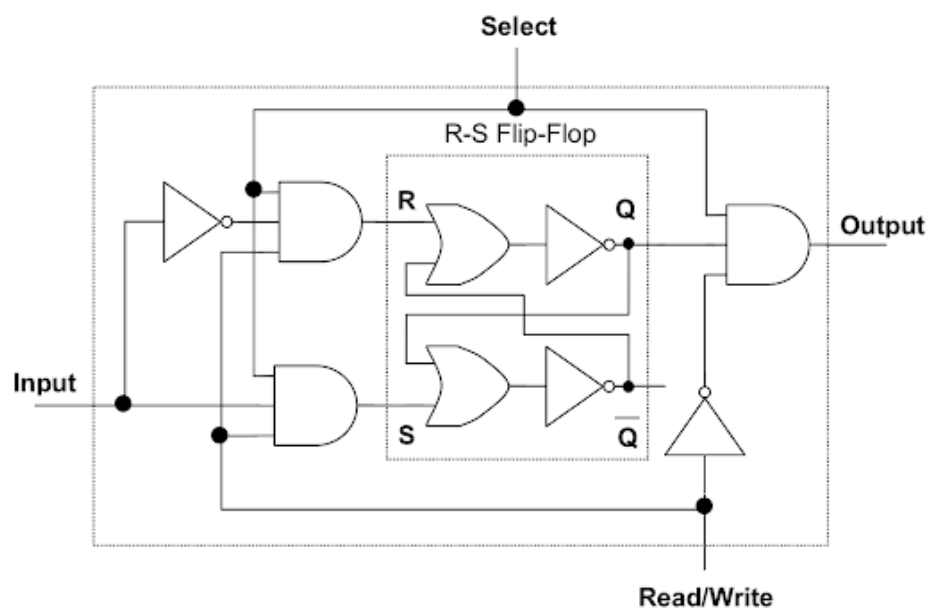
1.2.3.4 Памети

Паметите са най-сложните последователностни схеми, предназначени за съхранение на големи обеми данни. Съществуват различни типове памет: ROM (фиг. 1.41) за постоянно съхранение на данни, RAM (фиг. 1.42) за временно съхранение с произволен достъп, и EEPROM (фиг. 1.43) за препрограмируема постоянна памет. Всеки тип памет има своите специфични характеристики по отношение на време за достъп, капацитет и метод на програмиране.

[1]



Фиг. 1.41 Логическа схема на ROM



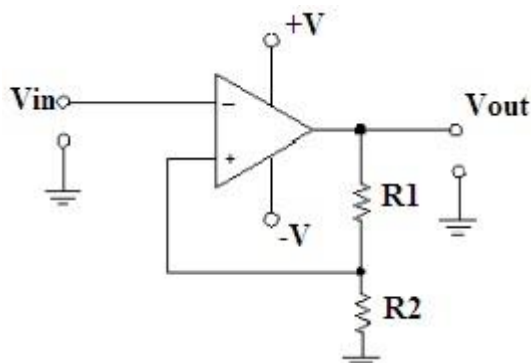
Фиг. 1.42 Логическа схема на RAM

1.2.4.2 Шум и допустими отклонения

Шумоустойчивостта е ключов параметър за надеждната работа на цифровите схеми. Тя се определя от разликата между изходните и входните логически нива, която създава т.нар. шумов резерв. Колкото по-голям е този резерв, толкова по-устойчива е схемата на външни смущения. ^{[7] [14] [23]}

1.2.4.3 Хистерезис

Хистерезисът е важна характеристика, особено при входни схеми, работещи с бавно променящи се сигнали. Той представлява разлика между праговете на превключване при нарастващ и спадащ сигнал, което значително повишава шумоустойчивостта на схемата. Тригерът на Шмит (фиг. 1.44) е класически пример за схема с хистерезис. ^[1]



Фиг. 1.44 Логическа схема на тригер на Шмит

1.2.4.4 Входни и изходни импеданси

Входните и изходните импеданси са критични параметри, определящи как схемите си взаимодействат. Високият входен импеданс минимизира натоварването на управляващата схема,

докато ниският изходен импеданс позволява управление на повече товари и по-дълги линии. [3]

1.2.4.5 Fan-in/Fan-out параметри

Fan-in и fan-out параметрите определят съответно колко входа може да има един логически елемент и колко входа на други елементи може да управлява един изход. Тези параметри са пряко свързани с натоварвателната способност и надеждността на схемите. [4]

1.2.5 Времеви параметри и закъснения

Времевите параметри са от ключово значение за правилното функциониране на цифровите схеми, особено при високи работни честоти. Те определят максималната скорост на работа и ограниченията при проектиране на сложни системи.

1.2.5.1 Време за разпространение на сигнала

Времето за разпространение на сигнала (propagation delay) е основен параметър, показващ закъснението между промяната на входния сигнал и съответната реакция на изхода. Това време зависи от използваната технология, броя на логическите нива в схемата и капацитивното натоварване на изхода. При съвременните технологии то е от порядъка на наносекунди, но може да стане значително при каскадно свързване на много елементи. [6] [11]

1.2.5.2 Време за нарастване и спадане

Времената за нарастване и спадане на сигнала показват колко време е необходимо на сигнала да премине от 10% до 90% от пълната амплитуда при нарастващ фронт и от 90% до 10% при спадащ фронт. Тези времена са особено важни при високочестотни приложения, тъй като определят максималната честота на превключване на схемата. При по-големи капацитетни товари тези времена се увеличават, което може да ограничи максималната работна честота.^[13]

1.2.5.3 Време на установяване и време на задържане

Времето за установяване (setup time) и времето за задържане (hold time) са критични параметри за последователностните схеми. Времето за установяване определя колко време преди активния фронт на тактовия сигнал входните данни трябва да бъдат стабилни, за да се гарантира правилното им запомняне. Времето за задържане показва колко време след тактовия импулс данните трябва да останат непроменени. Нарушаването на тези времеви изисквания може да доведе до метастабилни състояния и неправилна работа на схемата.^{[3] [4]}

1.2.5.4 Максимална работна честота

Максималната работна честота на цифровата схема се определя от най-дългия път на разпространение на сигнала между два последователни тригера. Този път включва времето за разпространение през логическите елементи, времената за нарастване и спадане на сигналите, както и времената за

установяване и задържане на приемащите тригери. Температурата и захранващото напрежение също влияят върху тези параметри и следователно върху максималната честота.^[10]

1.2.5.5 Времеви диаграми и анализ

Времевите диаграми са основен инструмент за анализ и документиране на работата на цифровите схеми. Те показват как сигналите се променят във времето и как си взаимодействат помежду си. Чрез тях могат да се открият потенциални проблеми като нарушения на времената за установяване и задържане, състезателни условия или недопустими комбинации от сигнали.^[9]

1.2.6 Натоварвателна способност

Натоварвателната способност на цифровите схеми е ключов фактор при проектирането на надеждни системи. Този параметър определя практическите граници на приложение на схемите и влияе върху тяхната производителност и надеждност.

1.2.6.1 Максимален изходен ток

Максималният изходен ток е параметър, определящ колко входа на други елементи може да управлява даден изход. При TTL технологията този ток типично е между 4 и 16 mA, докато при CMOS той е значително по-малък, но силно зависи от работната честота поради преобладаващия капацитетен характер на товара.^[24]

1.2.6.2 Брой входове, които могат да се управляват от един изход

Броят на входовете, които могат да бъдат управлявани от един изход (fan-out), се определя от съотношението между максималния изходен ток на управляващия елемент и входния ток на управляваните елементи. При надвишаване на този брой може да се наблюдава влошаване на логическите нива, увеличаване на времената за превключване или дори пълна загуба на функционалност.^[11]

1.2.6.3 Изходно съпротивление

Изходното съпротивление на цифровите схеми влияе както върху натоварвателната способност, така и върху бързодействието. По-ниското изходно съпротивление позволява по-големи изходни токове и по-бързо зареждане на капацитетните товари, но изисква по-мощни изходни каскади, което увеличава консумацията на енергия.^{[6] [14]}

1.2.6.4 Необходимост от буфериране

Буферирането става необходимо в случаите, когато трябва да се управляват много входове, дълги линии или товари с голям капацитет. Буферните елементи обикновено имат по-голяма натоварваща способност и по-малко изходно съпротивление, което им позволява да управляват по-големи товари без влошаване на параметрите на сигналите.^[4]

1.2.6.5 Влияние на капацитивен товар

Капацитивният товар има съществено влияние върху работата на цифровите схеми. Той увеличава времената за превключване, което води до намаляване на максималната работна честота. При големи капацитивни товари могат да възникнат нежелани осцилации на изходния сигнал, което налага използването на специални схеми за управление или допълнителни елементи за демпфериране. ^{[1] [3]}

1.2.6.6 Разсейвана мощност при различни натоварвания

Разсейваната мощност е важен параметър, особено при схеми с висока степен на интеграция. Тя се състои от статична компонента, дължаща се на токовете на утечка, и динамична компонента, свързана с превключването на елементите. Динамичната мощност нараства пропорционално на работната честота и капацитетния товар, което може да създаде сериозни проблеми с охлаждането при високочестотни приложения. ^{[11] [13]}

1.3 Проектиране на интегрални схеми

1.3.1 Методология за проектиране

Методологията за проектиране на интегрални схеми представлява систематичен подход към създаването на сложни електронни системи в рамките на един чип. Този процес започва с внимателно планиране и анализ на изискванията, които определят функционалността, производителността и ограниченията на бъдещата схема. Съвременната

методология се базира на концепцията за паралелно проектиране, където различни екипи работят едновременно върху различни аспекти на дизайна. В основата на методологията стои принципът на повторното използване на вече разработени и верифицирани компоненти. Това включва използването на стандартни клетки, IP блокове и параметризирани макроси, което значително ускорява процеса на разработка и намалява риска от грешки. Методологията също така поставя силен акцент върху автоматизацията на проектирането, използвайки специализирани CAD инструменти за симулация, оптимизация и верификация.

Важен аспект на методологията е интегрираният подход към управление на качеството. Това означава, че проверките за качество не се извършват само в края на процеса, а са вградени във всеки етап на разработката. Използват се различни методи за ранно откриване на проблеми, включително формална верификация, статичен анализ и симулации на различни нива на абстракция.^[1]

1.3.2 Йерархични нива на проектиране

Проектирането на интегрални схеми се осъществява на няколко йерархични нива, като всяко ниво представя системата с различна степен на абстракция. На най-високото ниво се намира системното проектиране, където се определят основните функционални блокове и взаимодействието между тях. Тук архитектите работят с абстрактни модели, фокусирайки се върху системните изисквания и общата структура на дизайна.

След системното ниво идва поведенческото проектиране, където се описва функционалността на всеки блок чрез езици за описание на хардуер като VHDL или Verilog. На това ниво се определят алгоритмите и протоколите за обработка на данни, без да се навлиза в детайлите на конкретната имплементация.

Следващото ниво е RTL, където дизайнът се описва чрез регистри и логически операции между тях. Това е критично ниво, тъй като то служи като мост между високото ниво на абстракция и конкретната хардуерна реализация. RTL дизайнът трябва да бъде синтезируем, което означава, че може да бъде автоматично преведен в схема от логически елементи.

На по-ниско ниво се намира логическото проектиране, където схемата се представя като мрежа от логически елементи и връзки между тях. Тук се извършва оптимизация на логическата структура с цел постигане на по-добра производителност и по-малка площ.

Най-ниското ниво е физическото проектиране, където логическата схема се трансформира в конкретно разположение на транзистори, проводници и други компоненти върху силициевия кристал. Това ниво изисква детайлно познаване на производствения процес и неговите ограничения. [6] [8] [11]

1.3.3 Етапи в проектирането на интегрални схеми

Процесът на проектиране на интегрални схеми преминава през няколко ключови етапа, всеки от които има специфични цели и резултати. Началният етап е спецификацията, където се дефинират изискванията към схемата, включително функционалност, производителност, консумация на енергия и ограничения за площ и цена. Това е критичен етап, тъй като грешки в спецификацията могат да доведат до сериозни проблеми в по-късните фази на проекта.

След спецификацията следва архитектурното проектиране, където се определя общата структура на системата. Тук се взимат важни решения относно разпределението на функционалността между хардуер и софтуер, избора на интерфейси и протоколи, както и стратегията за тестване и верификация.

RTL дизайнът е следващият ключов етап, където архитектурата се превръща в конкретен код на езиците VHDL или Verilog. Този код трябва

да бъде не само функционален, но и оптимизиран за синтез. На този етап се извършва и функционална верификация чрез симулация и формални методи.

Логическият синтез преобразува RTL кода в мрежа от логически елементи, оптимизирайки схемата за бързодействие, площ или консумация на енергия. След синтеза следва физическото проектиране, което включва няколко подетапа: разполагане на елементите, трасиране на връзките между тях и финални проверки за спазване на технологичните правила. ^[5]

1.3.4 Верификация и тестване

Верификацията и тестването са критични аспекти на процеса на проектиране на интегрални схеми, които гарантират коректността и надеждността на крайния продукт. Верификацията започва още на етапа на спецификация и продължава през целия процес на разработка. Тя включва различни методи и техники, адаптирани към различните нива на проектиране.

Функционалната верификация проверява дали дизайнът изпълнява правилно зададената спецификация. Това включва създаване на тестови сценарии, симулация на поведението на схемата и анализ на резултатите. Използват се различни подходи като насочено тестване, случайно генерирани тестове и формална верификация.

Статичната времева анализа проверява дали схемата ще работи правилно при зададената тактова честота, отчитайки закъсненията в логическите елементи и връзките между тях. Този анализ е особено важен за високочестотни схеми и трябва да се извършва при различни условия на околната среда.

Тестването за производство е друг важен аспект, който трябва да се планира още в началните етапи на проектиране. Това включва добавяне на

специални структури за тестване (DFT) и разработване на тестови вектори за производствения тест. [7] [14] [23]

1.3.5 Оптимизация на дизайна

Оптимизацията на дизайна е непрекъснат процес, който се извършва на всички нива на проектиране с цел подобряване на ключовите характеристики на схемата. Основните параметри, които се оптимизират, са бързодействие, площ и консумация на енергия, като често се налага търсене на компромис между тях.

На архитектурно ниво оптимизацията включва избор на подходящи алгоритми и структури за реализация на желаната функционалност. Това може да включва решения като използване на конвейерна обработка, паралелизъм или специализирани изчислителни блокове.

При RTL дизайна се прилагат различни техники за оптимизация като балансиране на критичните пътища, минимизиране на логическата дълбочина и ефективно използване на регистри. Важно е тези оптимизации да не нарушават функционалността на схемата и да са съвместими с процеса на синтез.

На ниво логически синтез се извършва автоматична оптимизация на схемата чрез различни алгоритми за минимизация на логическите функции и преподреждане на елементите. Тук е важно да се зададат правилни ограничения и цели за оптимизация, които отразяват реалните изисквания към схемата.

Физическата оптимизация включва техники като оптимално разполагане на елементите за минимизиране на дължината на връзките, буфериране на критични сигнали и балансиране на тактовата мрежа. Тези оптимизации трябва да се извършват при строго спазване на технологичните правила за производство.

Важен аспект на оптимизацията е и управлението на консумацията на енергия. Това включва техники като clock gating, power gating и

динамично управление на хранването. Тези техники стават все по-важни с увеличаването на сложността на схемите и изискванията за ниска консумация в мобилните устройства.

След извършване на оптимизациите е необходимо да се проведе пълна верификация, за да се гарантира, че промените не са нарушили функционалността или други важни параметри на схемата. Това често е итеративен процес, който продължава докато не се постигнат желаните характеристики при спазване на всички проектни ограничения. ^[4]

1.4. Работа със системата за автоматизирано проектиране Cadence Virtuoso

1.4.1 Въведение в Cadence Virtuoso

Cadence Virtuoso представлява мощна интегрирана среда за проектиране на интегрални схеми, която е се е утвърдила като индустриален стандарт в полупроводниковата индустрия. Тази система предоставя пълен набор от инструменти за целия процес на проектиране - от създаването на първоначалната схема до финалния layout и верификация на проекта. Virtuoso е изграден на базата на десетилетия опит в проектирането на интегрални схеми и непрекъснато се развива, за да отговори на нарастващите изисквания на съвременните полупроводникови технологии.

Системата е организирана в модулна архитектура, което позволява на различни екипи да работят едновременно върху различни аспекти на проекта. Всеки модул е специализиран за конкретна задача, като например аналогово проектиране, цифрово проектиране, физическо проектиране (layout) или верификация. Тази модулност осигурява гъвкавост и ефективност в работния процес, като същевременно поддържа тясна интеграция между различните етапи на проектиране.

Едно от ключовите предимства на Virtuoso е неговата способност да работи с различни технологични процеси и библиотеки. Системата поддържа широк спектър от производствени технологии, от зрели процеси

до най-съвременните нанометрови възли. Това се постига чрез използването на технологични файлове (PDK), които съдържат всички необходими параметри и правила за конкретния производствен процес.^[2]
[15]

1.4.2 Основни инструменти и функционалност

Основните инструменти в Cadence Virtuoso са организирани в интуитивен и логичен интерфейс, който позволява ефективна работа с различните аспекти на проектирането на интегрални схеми. Централно място заема Library Manager - инструмент за управление на библиотеки, клетки и изгледи. Той предоставя йерархичен изглед на проекта и позволява лесна навигация между различните компоненти.

1.4.2.1 Schematic Editor

Schematic Editor е основният инструмент за създаване и редактиране на електрически схеми. Той предлага богат набор от функции за работа със схеми, включително възможности за копиране, преместване и ротация на компоненти, автоматично свързване на проводници и създаване на йерархични дизайни. Редакторът поддържа параметризация на компонентите, което позволява лесна промяна на техните характеристики.^[2] [15]

1.4.2.2 Layout Editor

Layout Editor е специализиран инструмент за физическо проектиране на интегрални схеми. Той предоставя прецизен контрол върху геометрията на различните слоеве в интегралната схема и включва функции за автоматично генериране на често срещани структури.

Редакторът поддържа различни режими на изглед и мащабиране, което улеснява работата с комплексни дизайни. ^[15]

1.4.2.3 ADE

ADE е мощна среда за симулация и анализ на аналогови схеми. Тя интегрира различни видове анализи, включително DC, AC, transient и noise анализ. ADE позволява параметрични симулации и оптимизация на схемите, като предоставя богати възможности за визуализация и обработка на резултатите. ^{[1] [15]}

1.4.3 Създаване на схеми

Процесът на създаване на схеми в Cadence Virtuoso започва с отварянето на Schematic Editor и избора на подходяща библиотека и клетка. Интерфейсът на редактора е организиран така, че да предоставя бърз достъп до често използвани компоненти и функции. От лявата страна обикновено се намира палитра с компоненти, която може да се персонализира според нуждите на потребителя.

При поставяне на компоненти в схемата, системата автоматично поддържа електрическата свързаност и позволява лесно редактиране на връзките. Важна функционалност е възможността за параметризация на компонентите - всеки елемент може да има множество параметри, които определят неговото поведение в симулациите. Тези параметри могат да се променят директно в схемата или чрез специални диалогови прозорци.

Virtuoso поддържа създаването на йерархични дизайни, където сложни схеми могат да се разделят на по-малки, управляеми блокове. Това се постига чрез създаване на символи за схемите, които след това могат да се използват като компоненти в други схеми. Системата автоматично

поддържа консистентност между различните нива на йерархия и позволява лесна навигация между тях.

За улесняване на работата със схеми, Virtuoso предлага множество помощни функции като автоматично именуване на мрежи, проверка на свързаността, създаване на шини и работа с диференциални сигнали. Системата също така поддържа създаването на параметризирани генератори за често използвани схемни конфигурации.^[2]

1.4.4 Симулация и анализ

Симулацията и анализът в Cadence Virtuoso се осъществяват чрез ADE, който предоставя интегрирана среда за настройка, изпълнение и анализ на различни видове симулации. ADE поддържа множество симулатори, като най-често използваният е Spectre - мощен SPICE-базиран симулатор, оптимизиран за работа с аналогови и смесени схеми.

При настройката на симулация, потребителят може да избира между различни видове анализ:

- DC анализ за изследване на работната точка на схемата
- AC анализ за честотни характеристики
- Transient анализ за времеви характеристики
- Noise анализ за изследване на шумовите характеристики
- Monte Carlo анализ за изследване на влиянието на производствените толеранси
- Corner анализ за проверка на работата при различни условия

ADE предоставя богати възможности за визуализация на резултатите, включително създаване на графики, измервания и статистическа

обработка на данните. Резултатите могат да се експортират в различни формати за по-нататъшна обработка или документация. [2]

1.4.5 Проектиране на layout

Layout проектирането в Virtuoso се осъществява чрез специализиран Layout Editor, който предоставя прецизен контрол върху физическата реализация на интегралната схема. Редакторът работи с реални физически размери и поддържа всички необходими производствени слоеве, дефинирани в технологичния файл (PDK).

При създаване на layout, дизайнерът започва с позициониране на основните компоненти, следвайки определена стратегия за оптимизация на площта и минимизиране на паразитните ефекти. Virtuoso предоставя различни инструменти за прецизно позициониране и подравняване на елементите, както и функции за автоматично генериране на често срещани структури. [2]

Особено внимание се отделя на трасирането на връзките между компонентите. Layout Editor предлага както ръчно, така и полуавтоматично трасиране, като поддържа различни метални слоеве и автоматично създава необходимите контакти между тях. Системата следи в реално време за спазването на технологичните правила за минимални разстояния и широчини. [25]

1.4.6 DRC

DRC е критичен етап от верификацията на layout дизайна, който проверява дали физическата реализация отговаря на всички технологични правила, дефинирани от производствения процес. Virtuoso включва мощен DRC инструмент, който извършва автоматична проверка на целия дизайн.

DRC проверява множество аспекти на layout дизайна:

- Минимални широчини на линии и полигони
- Минимални разстояния между елементи
- Правила за плътност на металните слоеве
- Специални правила за определени структури
- Антена правила
- Правила за контакти и преходи между слоевете

При откриване на нарушения, системата генерира подробен отчет с местоположението и типа на всяко нарушение. Дизайнерът може интерактивно да навигира през нарушенията и да ги коригира директно в Layout Editor. ^[2]

1.4.7 LVS

LVS е процес на верификация, който проверява съответствието между електрическата схема и нейната физическа реализация в layout. LVS инструментът в Virtuoso извършва сложен анализ, който включва:

- Екстракция на електрическа схема от layout дизайна
- Сравнение на тази схема с оригиналната схема
- Проверка на съответствието на компонентите и техните параметри
- Верификация на свързаността между компонентите
- Проверка на размерите на транзисторите и други активни елементи

LVS процесът е особено важен, тъй като гарантира, че физическата реализация точно съответства на проектираната схема. При откриване на несъответствия, системата генерира подробен отчет, който помага за локализиране и отстраняване на проблемите.

Успешното преминаване на LVS проверката е задължително условие преди изпращане на дизайна за производство. След LVS може да се извърши и паразитна екстракция, която добавя към модела паразитните капацитети и съпротивления, въведени от физическата реализация. [2]

1.4.8 PEX

PEX (Parasitic Extraction) е важен етап от проектирането на интегрални схеми, който следва след успешното приключване на LVS верификацията. Този процес извлича паразитните елементи, които съществуват във физическата реализация на схемата, но не са видими в идеалния електрически модел.

Паразитната екстракция анализира физическата структура на layout дизайна и създава допълнителен електрически модел, който включва всички нежелани електрически ефекти. Тези ефекти възникват естествено поради близостта между различните слоеве и компоненти в интегралната схема.

Основните паразитни елементи, които се извличат при PEX анализа, включват:

- Паразитни капацитети между проводници на различни метални слоеве
- Паразитни съпротивления на металните връзки и контактите
- Капацитети към подложката
- Взаимни индуктивности между близко разположени проводници
- Свързващи капацитети между съседни метални линии

PEX процесът е изключително важен за точното симулиране на поведението на схемата при високи честоти, където паразитните ефекти могат значително да повлияят на работата ѝ. Екстрахираният модел позволява на проектантите да:

- Оценят влиянието на паразитните ефекти върху бързодействието на схемата
- Откриват потенциални проблеми със закъснения на сигналите
- Анализират връзката между layout структурата и електрическите характеристики
- Оптимизират дизайна за минимизиране на нежеланите ефекти

След извършване на паразитната екстракция, получените данни се използват за по-прецизни симулации, които отчитат реалното поведение на схемата в силициевата подложка. Това е особено критично за високочестотни схеми, аналогови блокове и схеми с висока степен на интеграция, където паразитните ефекти могат да доведат до значителни отклонения от идеалното поведение.

1.5. Process Design Kit

1.5.1 Структура на PDK

Process Design Kit представлява основата на проектирането на интегрални схеми, служейки като ключов интерфейс между проектанта на схемата и полупроводниковия производствен процес. В своята същност PDK е внимателно организирана колекция от файлове и документация, която позволява на проектантите да създават интегрални схеми, които могат успешно да бъдат произведени използвайки специфична полупроводникова производствена технология.

Структурата на PDK е йерархично организирана, за да улесни ефективния достъп до различните компоненти и информация. На най-високо ниво обикновено съдържа няколко основни директории, всяка от които обслужва специфична цел в процеса на проектиране. Основните директории обикновено включват технологични файлове, модели на

устройства, библиотеки със стандартни клетки, правила за физическа верификация и параметри за екстракция на паразитни елементи.

Организацията на тези файлове следва логическа структура, която отразява процеса на проектиране, започвайки от базовите технологични дефиниции и стигайки до подготовката за производство. Например, технологичната директория съдържа фундаментална информация за процеса като дефиниции на слоевете, физически константи и спецификации на производствената решетка. Директорията с модели съдържа всички модели на устройства, организирани по тип (MOS, биполарни, пасивни компоненти), докато директорията за верификация съдържа файлове с правила за проверка на дизайна (DRC) и съответствие между схема и топология (LVS).

Един от най-критичните аспекти на структурата на PDK е контролът на версиите и управлението на конфигурацията. Всяка версия на PDK трябва да бъде внимателно проследявана и документирана, тъй като дори малки промени могат значително да повлияят на резултатите от проектирането. Системата за контрол на версиите обикновено включва подробна документация на промените между версиите, информация за съвместимост и всички известни проблеми или ограничения. ^{[5] [12]}

1.5.2 Компоненти и модели

Секцията за компоненти и модели в PDK представлява градивните елементи, достъпни за проектантите в дадена технология. Това включва както активни устройства (като транзистори), така и пасивни компоненти (резистори, кондензатори, индуктори), всеки характеризирани с подробни електрически модели, които точно представят тяхното поведение при различни работни условия.

За MOSFET устройствата моделите обикновено включват различни варианти, оптимизирани за специфични цели, като високоскоростна

работа, ниска консумация на енергия или толерантност към високо напрежение. Всеки модел обхваща поведението на устройството в различни работни режими, включително ефекти като насищане на скоростта, модулация на дължината на канала и различни ефекти при къс канал, които стават значими в съвременните технологични възли.

Пасивните компоненти са моделирани със същата прецизност, включвайки техните неидеални характеристики. Например, моделите на резисторите включват температурни коефициенти и зависимости от напрежението, докато моделите на кондензаторите отчитат зависимата от напрежението капацитивност и паразитни ефекти. Моделите на индукторите са особено сложни, включвайки ефекти на взаимна индукция и честотно-зависимо поведение.

Точността на тези модели е от решаващо значение за успеха на проектирането. Всеки модел преминава през обширна валидация спрямо реални измервания върху силиций при различни технологични ъгли, температури и работни условия. Данните от валидацията често са включени в документацията на PDK, помагайки на проектантите да разберат ограниченията и точността на моделите. [5] [8]

1.5.3 Технологични библиотеки

Технологичните библиотеки в PDK представляват предварително проектирани и валидирани схемни елементи, които проектантите могат да използват като градивни блокове за по-големи проекти. Тези библиотеки обикновено включват стандартни клетки (базови логически елементи, тригери, латчове), входно/изходни клетки (I/O падове) и понякога по-сложни функционални блокове.

Библиотеката със стандартни клетки е особено важна, тъй като формира основата за цифрово проектиране. Всяка клетка в библиотеката е внимателно проектирана да отговаря на специфични критерии за

производителност, като същевременно поддържа съвместимост с инструментите за автоматично разполагане и опроводяване. Клетките са характеризирани при различни работни условия, предоставяйки информация за време закъснение, консумация на мощност и шум, която позволява точна симулация и оптимизация на схемата.

Процесът на характеризация за тези библиотеки е обширен, създавайки подробни времеви модели, които обхващат поведението на клетките при различни условия. Това включва времена за установяване и задържане, времена на разпространение и консумация на мощност при различни технологични ъгли, напрежения и температури (PVT ъгли). Получените данни обикновено се предоставят в индустриални стандартни формати като Liberty (.lib) файлове за времеви анализ и LEF файлове за физическо проектиране.

I/O библиотеките изискват специално внимание, тъй като те осъществяват интерфейс между напрежителната област на ядрото на чипа и външните напрежения. Тези клетки включват сложни защитни схеми и възможности за преобразуване на нивата, всички внимателно характеризирани, за да осигурят надеждна работа при различни условия, включително събития на електростатично разреждане (ESD).^{[5][12]}

1.5.4 Design rules

Правилата за проектиране формират критичен компонент на PDK, дефинирайки производствените ограничения, които трябва да бъдат спазвани, за да се гарантира успешно производство на чипа. Тези правила произтичат от ограниченията на производствения процес и са от съществено значение за постигане на приемливи нива на добив.

Правилата обхващат различни аспекти на проектирането на топологията, включително минимални размери на елементите, изисквания за разстояния, правила за плътност и специални производствени изисквания.

Например, правилата за металните слоеве определят минимална ширина и разстояния, правилата за виите дефинират как различните метални слоеве могат да бъдат свързвани, а правилата за антени защитават устройствата от повреди по време на производствения процес.

Съвременните правила за проектиране са станали все по-сложни с напредналите технологични възли. Те сега включват контекстно-зависими правила, където изискванията се променят въз основа на околната геометрия, както и сложни изисквания за плътност и равномерност, които гарантират, че химико-механичното полиране (CMP) може да постигне необходимата планарност.

Наборът от правила обикновено се предоставя във формат, съвместим с основните EDA инструменти, често включвайки както версия, четима от човек, така и специфични за инструментите имплементации. Правилата са придружени от подробна документация, обясняваща тяхната цел и предоставяща примери за правилни и неправилни имплементации. ^{[5] [8]}

1.5.5 Екстракция на паразити

Параметрите за екстракция в PDK позволяват точно предвиждане на паразитните ефекти, които влияят върху работата на схемата. Тези параметри позволяват на инструментите да извличат стойности на съпротивление, капацитет и индуктивност от физическата топология, които са от решаващо значение за точна симулация на схемата и финален времеви анализ.

Процесът на екстракция отчита различни физически ефекти, включително периферен капацитет между съседни метални линии, капацитет на свързване между различни метални слоеве и разпределено съпротивление в междусъединенията. Съвременните PDK включват също модели за ъглови ефекти, зависим от ширината капацитет и различни вариации, свързани с производството.

Точността на параметрите за екстракция става все по-критична в съвременните възли, където паразитните ефекти могат да доминират производителността на схемата. PDK обикновено предоставя различни нива на точност на екстракцията, от бързи приблизителни методи за първоначални итерации на проектиране до високоточни методи за финално одобрение.

Параметрите за екстракция редовно се калибрират спрямо реални измервания върху силиций, за да се гарантира тяхната точност. Този процес на калибриране включва тестови структури, специално проектирани за изолиране и измерване на различни паразитни ефекти, като резултатите се използват за усъвършенстване на моделите за екстракция. [5] [12]

1.5.6 Интеграция с Cadence среда

Интеграцията на PDK със средата Cadence е от решаващо значение за осигуряване на ефективни работни процеси при проектиране. Тази интеграция включва настройка на PDK за безпроблемна работа с различни инструменти на Cadence, включително Virtuoso за потребителско проектиране, Innovus за цифрова имплементация и различни инструменти за верификация.

Процесът на интеграция включва няколко ключови аспекта. Първо, технологичният файл трябва да бъде правилно конфигуриран, за да дефинира наличните слоеве, проектната решетка и други базови параметри в средата Cadence. Това включва настройка на ресурси за визуализация, които дефинират как различните слоеве се появяват в редактора на топология, и конфигуриране на предпочитани посоки за опроводяване за автоматизирани инструменти.

Настройката на библиотеки и клетки е друг критичен аспект, осигуряващ правилното разпознаване и възможност за инстанциране на всички

стандартни клетки и други предварително проектирани компоненти. Това включва конфигуриране на символни и топологични представления, както и осигуряване на правилно свързване на всички необходими файлове с модели за симулация.

Интеграцията включва също настройка на процеси за верификация, конфигуриране на инструменти като Assura или Pegasus за физическа верификация и установяване на връзки с инструменти за екстракция и симулация. Това често включва създаване на специфични за технологията файлове с правила и скриптове, които автоматизират често срещани задачи при проектиране.

Често се добавят потребителски менюта и функции за автоматизация за повишаване на продуктивността, като например генератори на параметризирани клетки (PCells), които могат да създават инстанции на топологии с дефинирани от потребителя параметри. Тези функции трябва да бъдат внимателно тествани, за да се гарантира, че работят правилно в средата Cadence.

Интеграцията на PDK с Cadence трябва също да обработва механизми за контрол на версиите и актуализации, осигурявайки възможност проектантите лесно да превключват между различни версии на PDK, като същевременно поддържат консистентност на проекта. Това включва управление на технологични библиотеки, файлове с модели и набори от правила за верификация по начин, който предотвратява несъответствия между версиите. [5] [27]

ВТОРА ГЛАВА

Принцип на работа на 74НС42

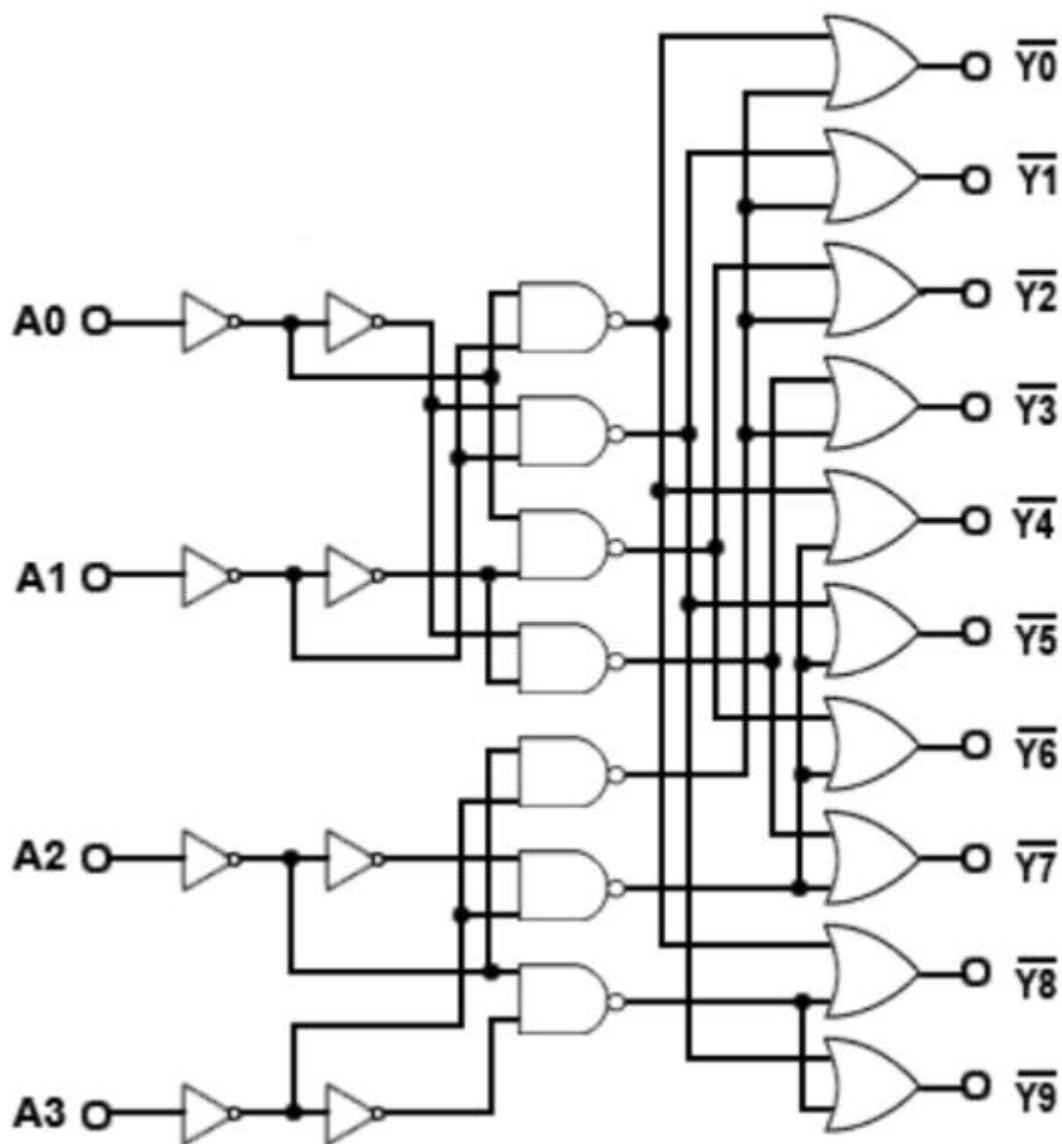
Интегралната схема 74НС42 (*фиг. 2.1*) представлява високоскоростен CMOS BCD към десетичен декодер, чийто принцип на работа се основава на преобразуването на четирибитов двоичен код в десетичен изход чрез специфична логическа структура. Схемата приема четири входни сигнала (A0, A1, A2 и A3), които формират BCD код, и генерира десет изходни сигнала (Y0 до Y9), активни при ниско логическо ниво. (*фиг. 2.2*)

Входният BCD код представлява двоично кодирано десетично число, като валидните входни комбинации са ограничени от 0000 до 1001, съответстващи на десетичните числа от 0 до 9. При постъпване на входен сигнал, първият етап на обработка включва буфериране чрез входни инвертори, които осигуряват стабилност на входните нива и защита от смущения. След буферирането сигналите преминават през комбинационна логическа матрица, състояща се от NAND и OR елементи, които извършват същинското декодиране.

Основният декодиращ механизъм функционира чрез специфична комбинация от логически елементи, където всеки изход се активира от уникална комбинация на входните сигнали. Когато определена входна комбинация бъде разпозната, съответният изход преминава в ниско логическо ниво (логическа 0), докато всички останали изходи се поддържат във високо ниво (логическа 1). Този принцип гарантира, че във всеки момент само един изход може да бъде активен, което е критично за правилното функциониране на схемата в приложения като управление на дисплеи или адресна декодировка.

Схемата притежава вграден защитен механизъм за обработка на невалидни входни комбинации. При постъпване на BCD код, по-голям от 1001 (десетично 9), логическата структура разпознава това като невалидно състояние и реагира чрез деактивиране на всички изходи, като ги установява във високо логическо

ниво. Този механизъм предотвратява генерирането на некоректни изходни сигнали и осигурява надеждна работа в реални приложения.



Фиг. 2.1 Логическа схема на 74НС42

INPUTS				OUTPUTS									
A3	A2	A1	A0	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9
0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1	1	1
0	1	1	0	1	1	1	1	1	1	0	1	1	1
0	1	1	1	1	1	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0
1	0	1	0	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1

Фиг. 2.2 Таблица на истинност на 74НС42

ТРЕТА ГЛАВА

Практическа реализация и анализ на логически елементи

В настоящата глава са представени резултатите от проектирането, симулацията и физическата реализация на трите логически елемента - OR, NAND и NOT. Разгледани са всички етапи на проектирането, включително схемотехническа реализация, DC анализ за оразмеряване на транзисторите, времеви анализ за верификация на функционалността и физическа реализация с последващи DRC и LVS проверки.

3.1 Проектиране на основни логически елементи

3.1.1 Реализация на OR елемент

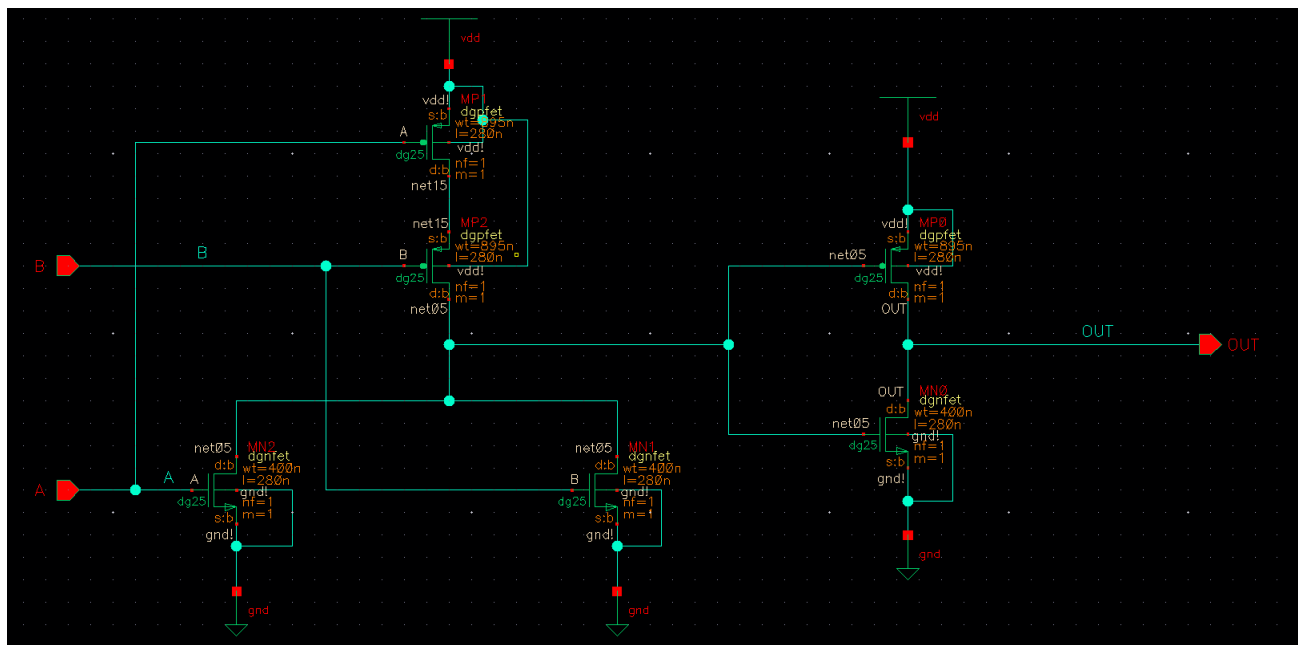
3.1.1.1 Схемно представяне

Схемното решение на OR елемента (*фиг. 3.1*) е реализирано с използването на CMOS технология. Схемата се състои от NAND логически елемент, свързан последователно с NOT логически елемент.

NAND логическият елемент се състои от два PMOS транзистора, свързани последователно в горната част, и два NMOS транзистора, свързани паралелно в долната част. Входните сигнали А и В управляват съответните двойки PMOS/NMOS транзистори.

Изходът на NAND елемента е свързан с входа на NOT логическия елемент. NOT елементът се състои от един PMOS транзистор и един NMOS транзистор.

Така реализираната схема функционално представлява OR логически елемент - когато поне един от входните сигнали А или В е висок, на изхода ще се получи висок сигнал.



Фиг. 3.1 Схемно представяне на OR

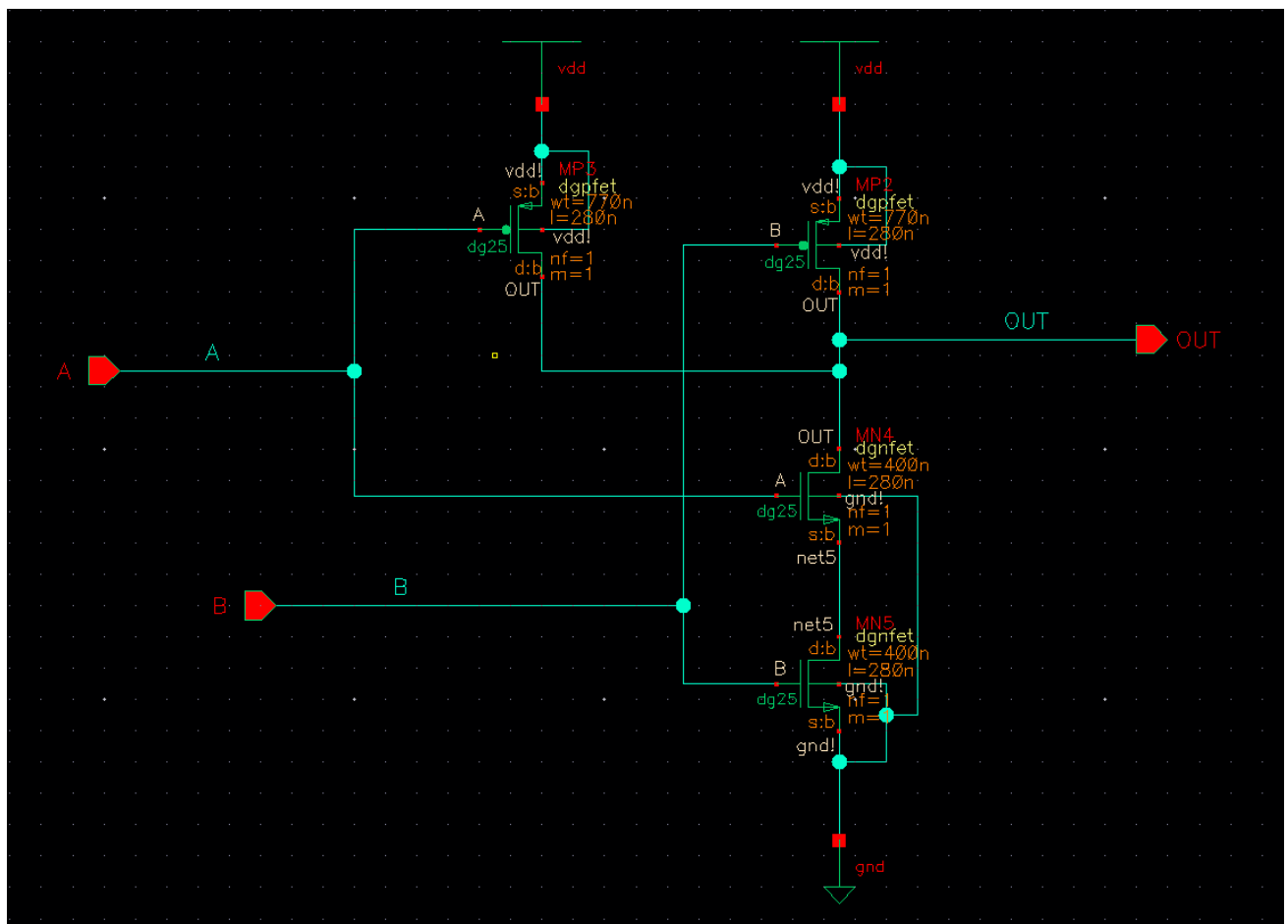
3.1.1.2 Описание на функционалността

Функционално, елементът реализира логическа операция ИЛИ, като изходът е във високо ниво (2.5V) когато поне един от входовете е във високо ниво. Изходът е в ниско ниво (0V) само когато и двата входа са в ниско ниво.

3.1.2 Реализация на NAND елемент

3.1.2.1 Схемно представяне

Схемното решение на NAND елемента (фиг. 3.2) е реализирано с използването на CMOS технология. Схемата се състои от два PMOS транзистора, свързани последователно в горната част, и два NMOS транзистора, свързани паралелно в долната част. Входните сигнали А и В управляват съответните двойки PMOS/NMOS транзистори.



Фиг. 3.2 Схемно представяне на NAND

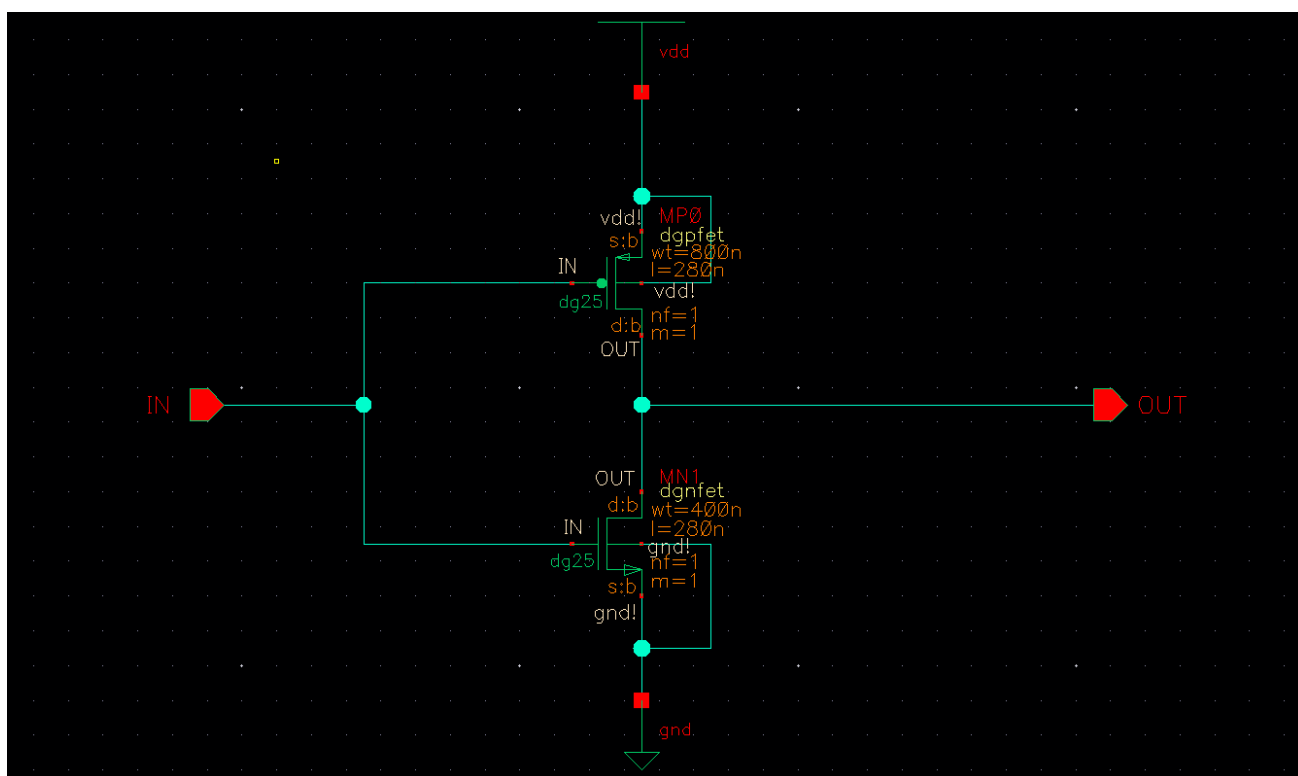
3.1.2.2. Описание на функционалността

Елементът реализира логическата операция И-НЕ, при която изходът е в ниско ниво (0V) само когато и двата входа са във високо ниво (2.5V). Във всички останали случаи изходът е във високо ниво.

3.1.3 Реализация на NOT елемент

3.1.3.1 Схемно представяне

NOT елементът е реализиран чрез CMOS технология (фиг. 3.3), използвайки един PMOS и един NMOS транзистор. Входният сигнал IN управлява едновременно двата транзистора по комплементарен начин.



Фиг. 3.3 Схемно представяне на NOT

3.1.3.2 Описание на функционалността

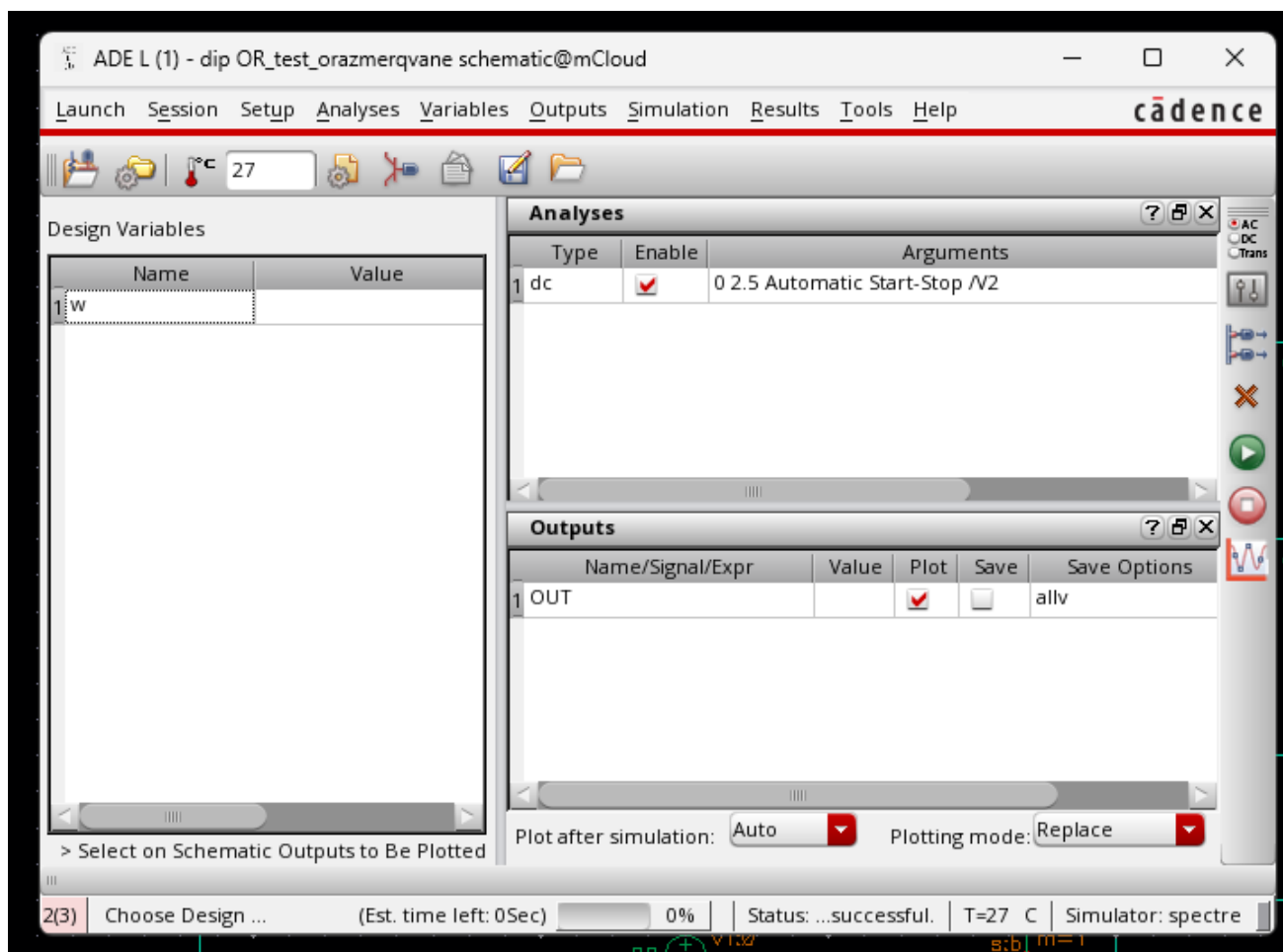
Елементът реализира логическата операция НЕ, при която изходът е в противоположно състояние спрямо входа. Когато входът е във високо ниво (2.5V), изходът е в ниско ниво (0V) и обратно.

3.2 DC анализ и оразмеряване на логическите елементи

3.2.1 Методология за DC анализ

За извършване на DC анализа е използвана средата Cadence Virtuoso ADE L (*фиг. 3.4*). Анализът е проведен чрез променяне на входното напрежение от 0V до 2.5V и наблюдение на изходното напрежение. За всички елементи са използвани следните параметри:

- Захранващо напрежение: 2.5V
- Температура: 27°C
- Стъпка на анализа: 0.01V



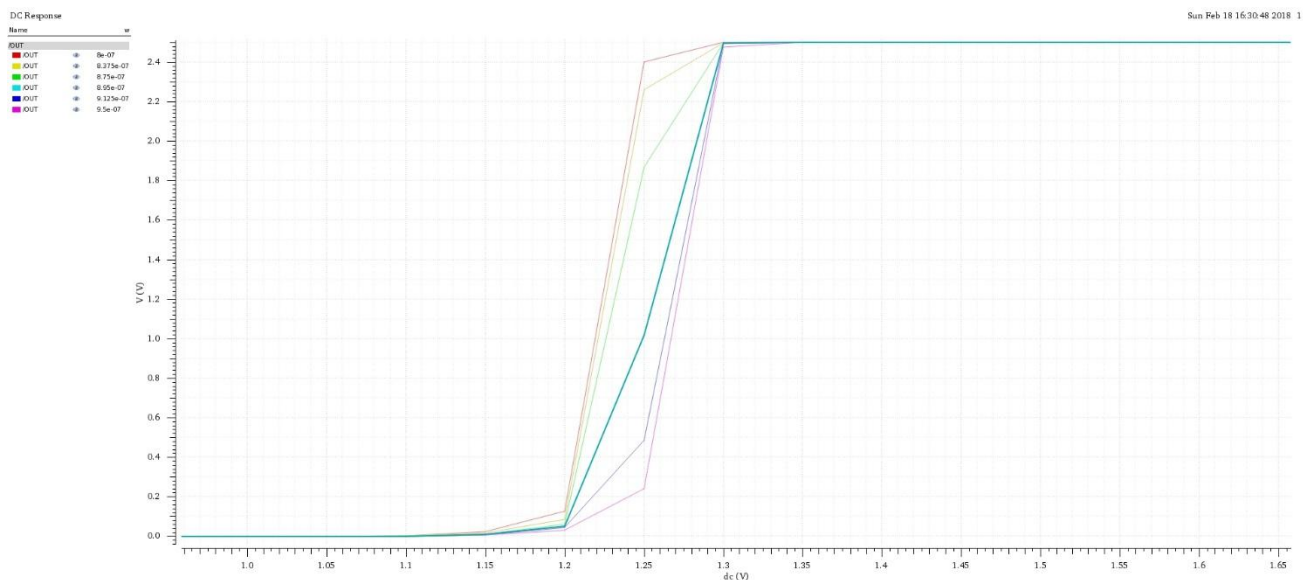
Фиг. 3.4 DC анализ чрез Cadence Virtuoso ADE L

3.2.2 DC анализ и оразмеряване на OR елемент

На базата на DC анализа (фиг. 3.5) са определени оптималните размери на транзисторите:

- PMOS: W=895nm, L=280nm
- NMOS: W=400nm, L=280nm

Съотношението между размерите на PMOS и NMOS транзисторите е оптимизирано за постигане на симетрична характеристика на превключване.



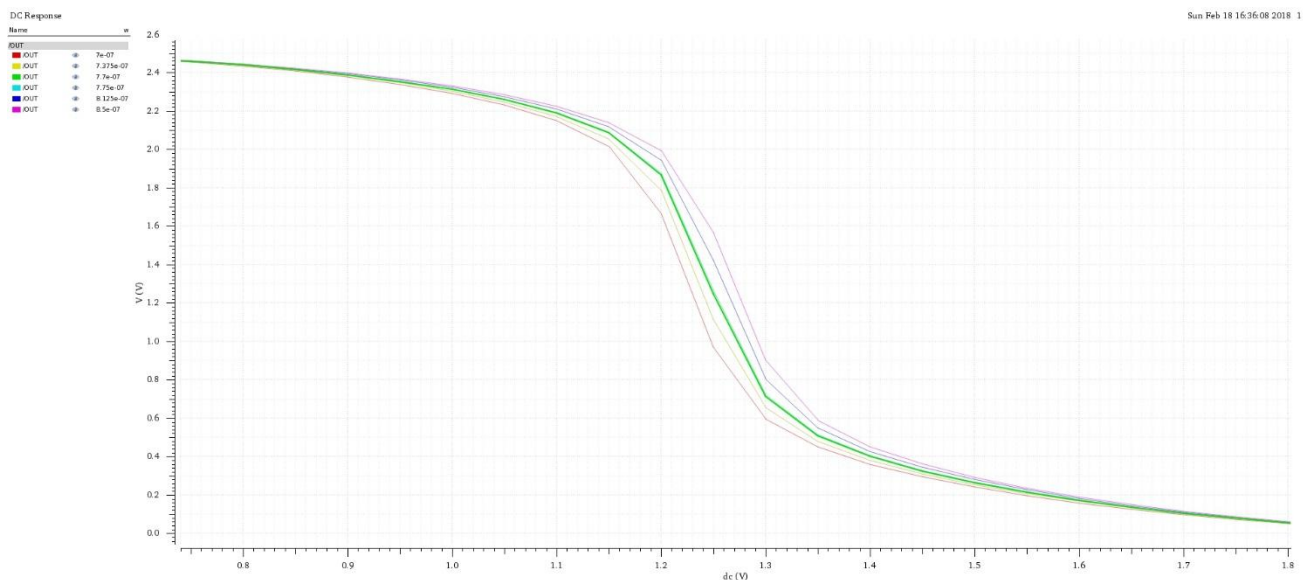
Фиг. 3.5 DC анализ на OR

3.2.3 DC анализ и оразмеряване на NAND елемент

На базата на DC анализа (фиг. 3.6) са определени оптималните размери на транзисторите:

- PMOS: W=770nm, L=280nm
- NMOS: W=400nm, L=280nm

Съотношението между размерите на PMOS и NMOS транзисторите е оптимизирано за постигане на симетрична характеристика на превключване.



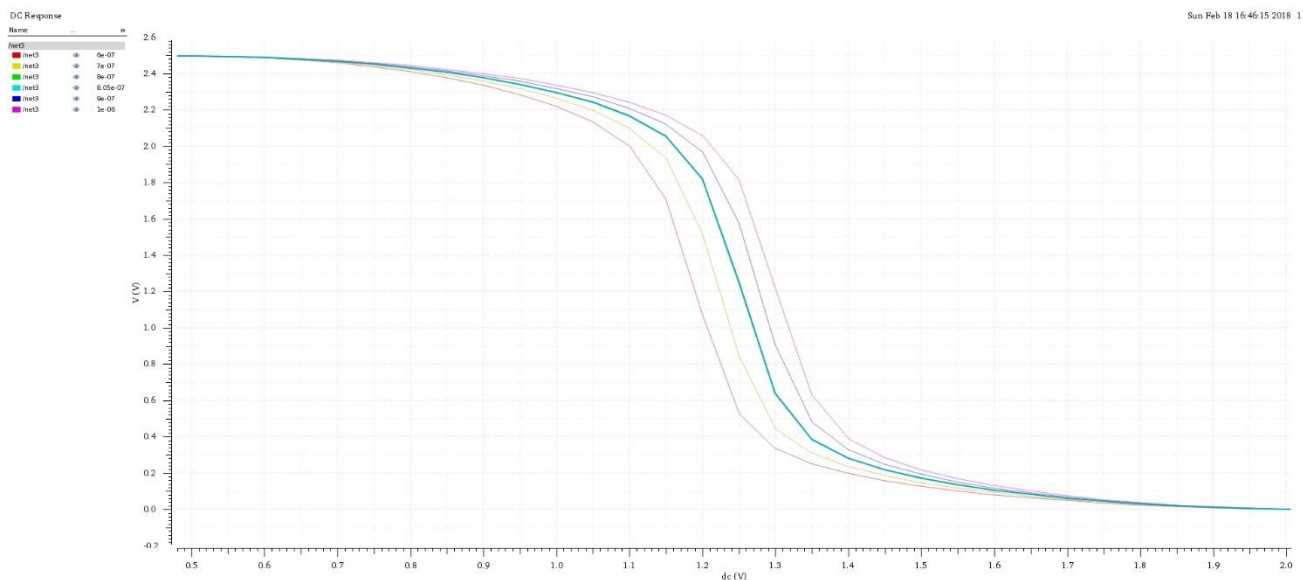
Фиг. 3.6 DC анализ на NAND

3.2.4 DC анализ и оразмеряване на NOT елемент

На базата на DC анализа (фиг. 3.7) са определени оптималните размери на транзисторите:

- PMOS: W=805nm, L=280nm
- NMOS: W=400nm, L=280nm

Съотношението между размерите на PMOS и NMOS транзисторите е оптимизирано за постигане на симетрична характеристика на превключване.



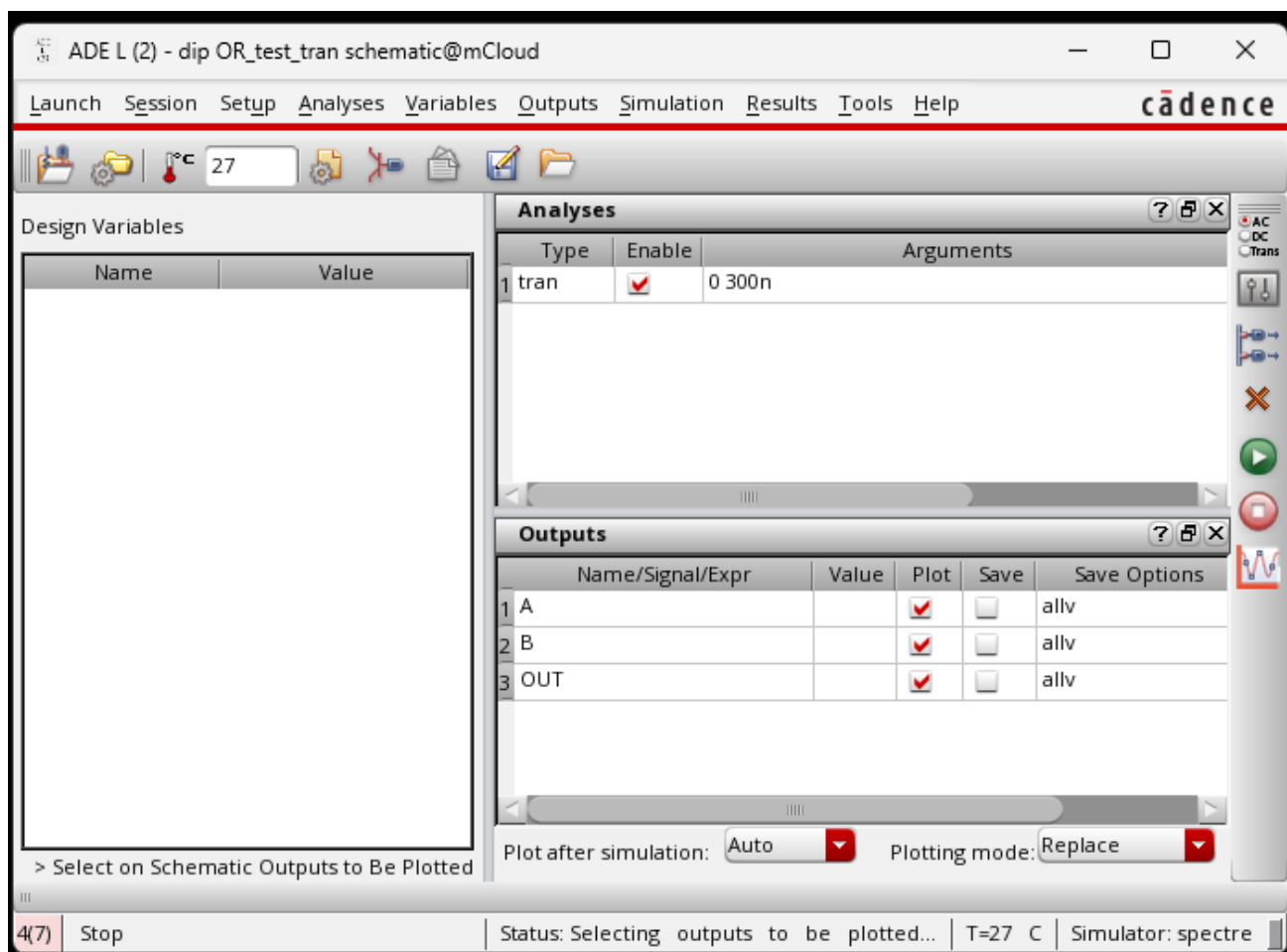
Фиг. 3.7 DC анализ на NOT

3.3 Времеви (Transient) анализ на логическите елементи

3.3.1 Методология за времеви анализ

Времевият анализ е проведен в ADE L (фиг. 3.8) със следните параметри:

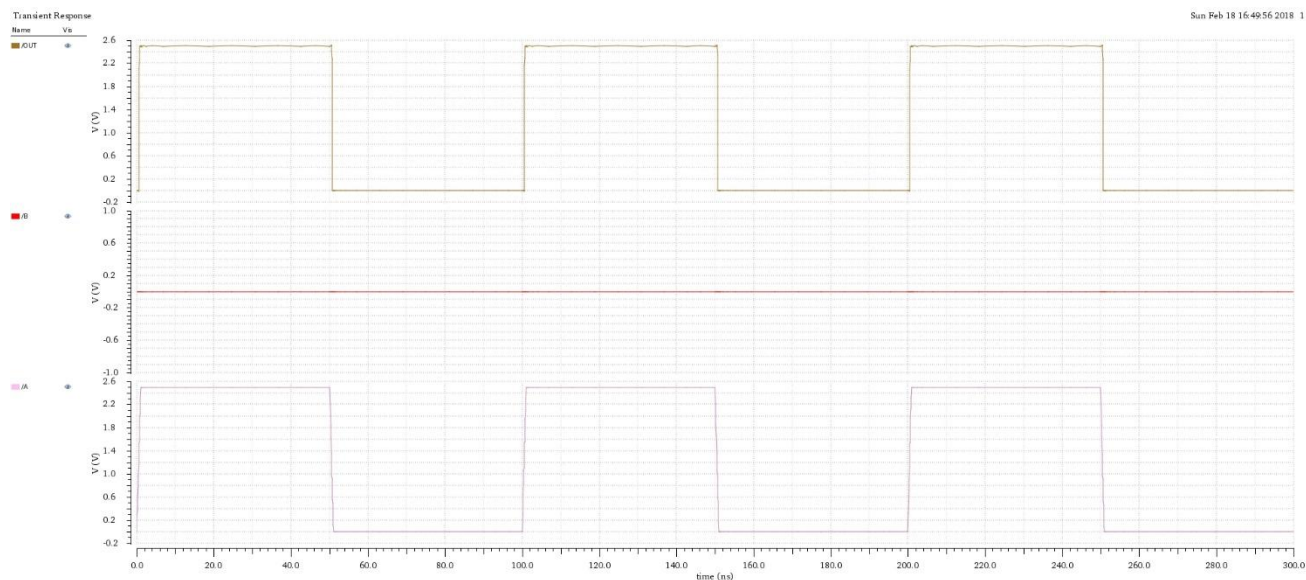
- Време на симулация: 300ns
- Стъпка: 0.1ns
- Входни сигнали: правоъгълни импулси с период 100ns
- Захранващо напрежение: 2.5V



Фиг. 3.8 Времеви анализ чрез Cadence Virtuoso ADE L

3.3.2 Времеви анализ на OR елемент

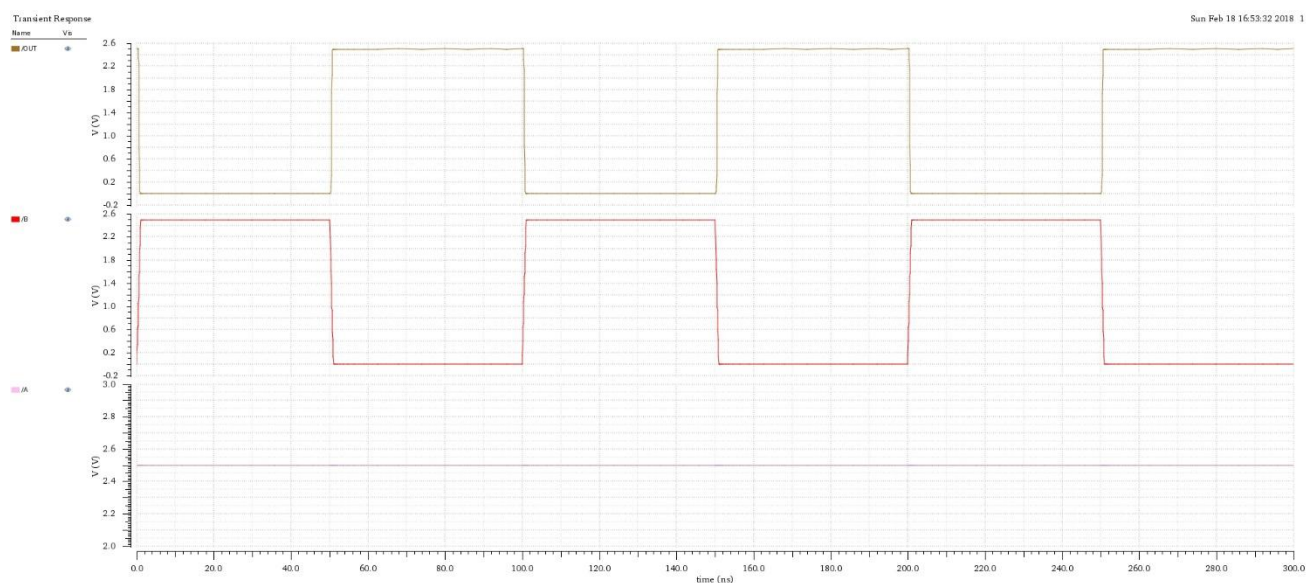
Времевата диаграма (фиг. 3.9) потвърждава правилната работа на OR елемента според таблицата на истинност. Анализът показва време на нарастване около 0.5ns и време на спадане около 0.4ns.



Фиг. 3.9 Времеви анализ на OR

3.3.3 Времеви анализ на NAND елемент

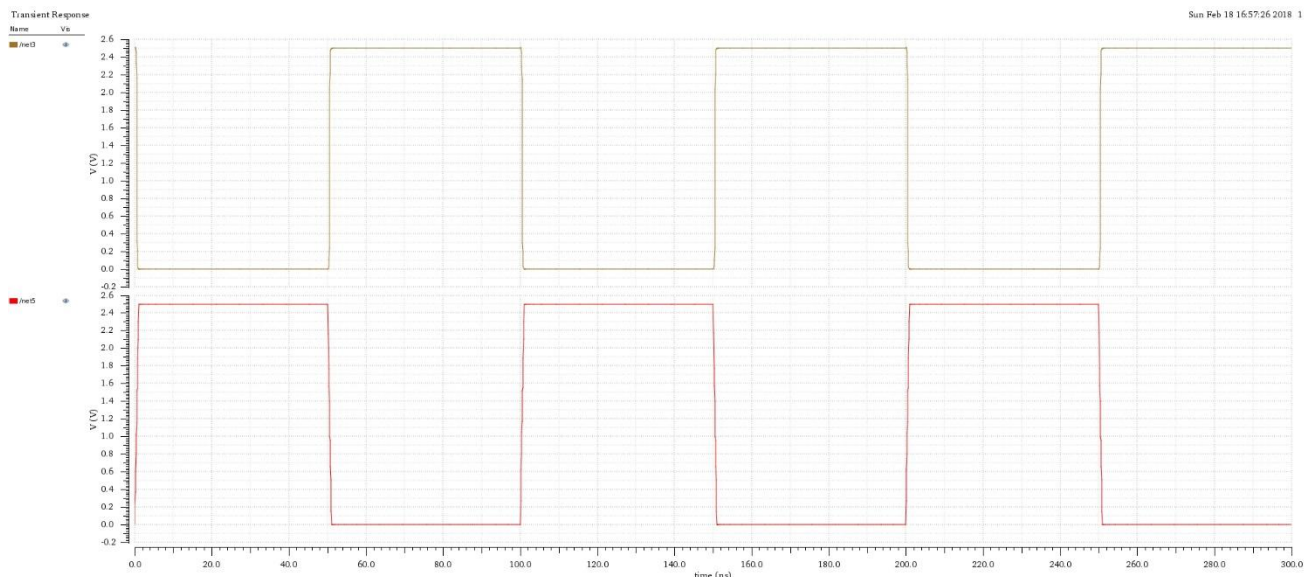
Времевият анализ на NAND елемента (фиг. 3.10) демонстрира коректно функциониране при всички комбинации от входни сигнали. Времената на превключване са съизмерими с тези на OR елемента.



Фиг. 3.10 Времеви анализ на NAND

3.3.4 Времеви анализ на NOT елемент

Времевата диаграма на NOT елемента (фиг. 3.11) показва правилно инвертиране на входния сигнал с минимално закъснение около 0.3ns.



Фиг. 3.11 Времеви анализ на NOT

3.4 Проектиране на физическа реализация (Layout)

3.4.1 Методология за проектиране на layout

При проектирането на layout са използвани следните слоеве(фиг.3.12):

За дифузионните области (p+ и n+):

PX и NW слоевете служат за определяне на дифузионните области. PX слойт дефинира активните области за p+ дифузия, докато NW слойт определя областите за n-кладенци, където ще се формират n+ дифузионни области.

За поликристалния силиций:

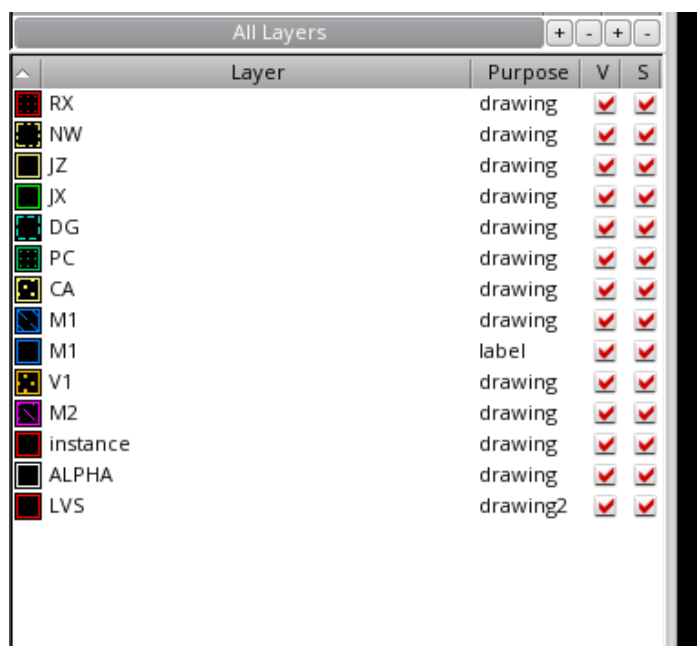
PC слойт, който определя областите с поликристален силиций, който служи за гейтовете на транзисторите. Този слой е критичен за определяне на електрическите характеристики на устройството.

За металните слоеве:

M1 е първият метален слой, който се използва за локални връзки и захранване. M2 (M2) е вторият метален слой, който обикновено се използва за по-дълги връзки и глобално разпределение на сигналите.

За контактите и пробивните контакти:

CA (слоят определя контактите между първия метален слой (M1) и поликристалния силиций или дифузионните области. V1 слой дефинира пробивните контакти (via) между първия (M1) и втория (M2) метален слой.

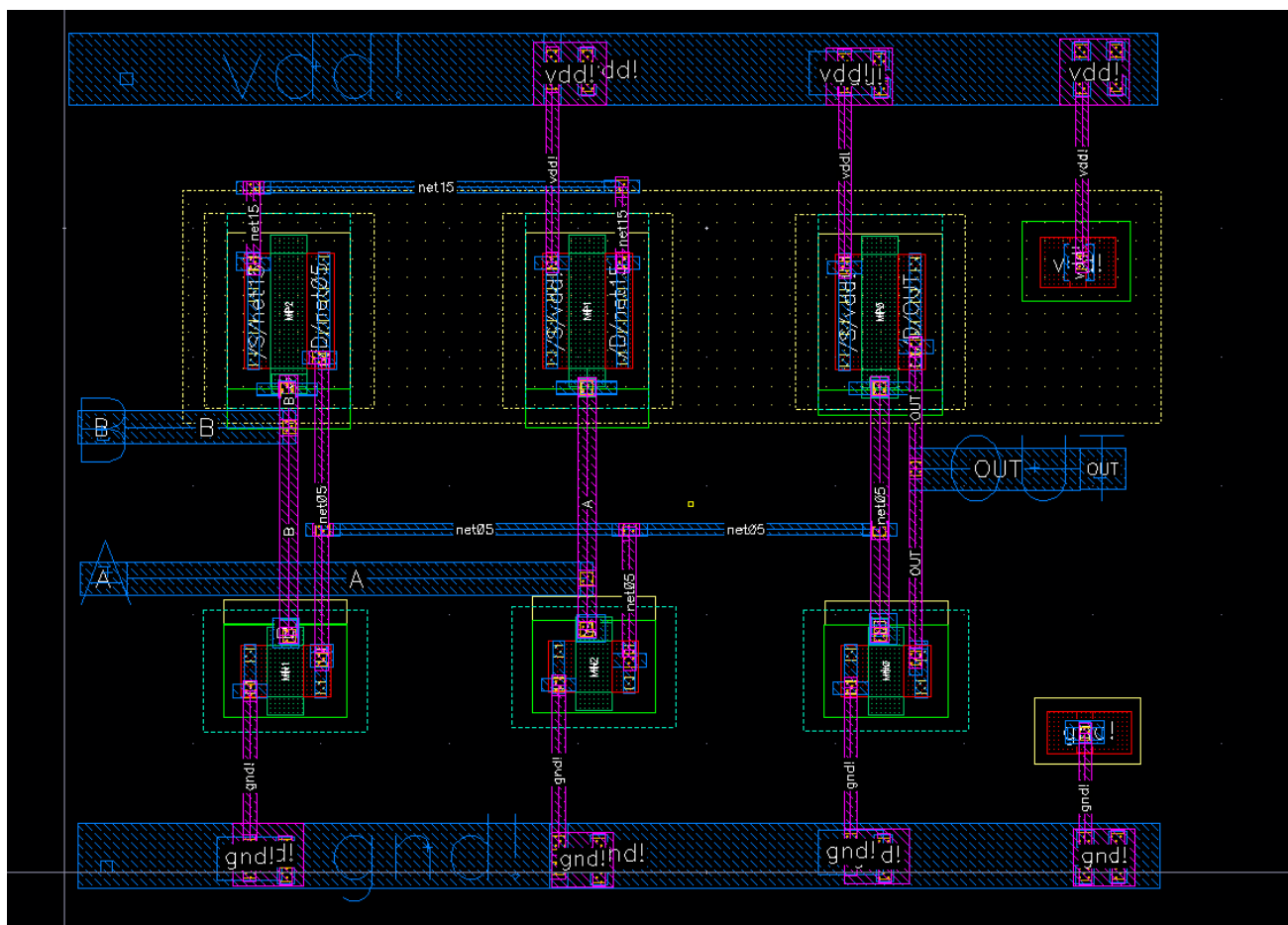


All Layers				
	Layer	Purpose	V	S
	RX	drawing	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	NW	drawing	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	JZ	drawing	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	JX	drawing	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	DG	drawing	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	PC	drawing	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	CA	drawing	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	M1	drawing	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	M1	label	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	V1	drawing	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	M2	drawing	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Instance	drawing	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	ALPHA	drawing	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	LVS	drawing2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Фиг. 3.12 Използвани слоеве при проектирането на layout

3.4.2 Layout на OR елемент

Физическата реализация на OR елемента (фиг. 3.13) е проектирана с минимални размери съгласно технологичните правила. DRC проверките са проведени успешно, без установени нарушения (фиг. 3.14). LVS верификацията потвърждава пълно съответствие между схема и layout (фиг. 3.15).



Фиг. 3.13 Физическа реализация на OR

```

Total CPU Time           : 4(s)
Total Real Time          : 4(s)
Peak Memory Used         : 44(M)
Total Original Geometry  : 193(394)
Total DRC RuleChecks     : 2260
Total DRC Results        : 0 (0)
Summary can be found in file OR.sum
ASCII report database is /home/chavdara_beleva/Diplomna/diplomna/DRC/OR/OR.drc_errors.ascii
Checking in all SoftShare licenses.

Design Rule Check Finished Normally. Wed Jan 31 00:30:02 2018

```

Фиг. 3.14 Резултат от DRC на OR

```

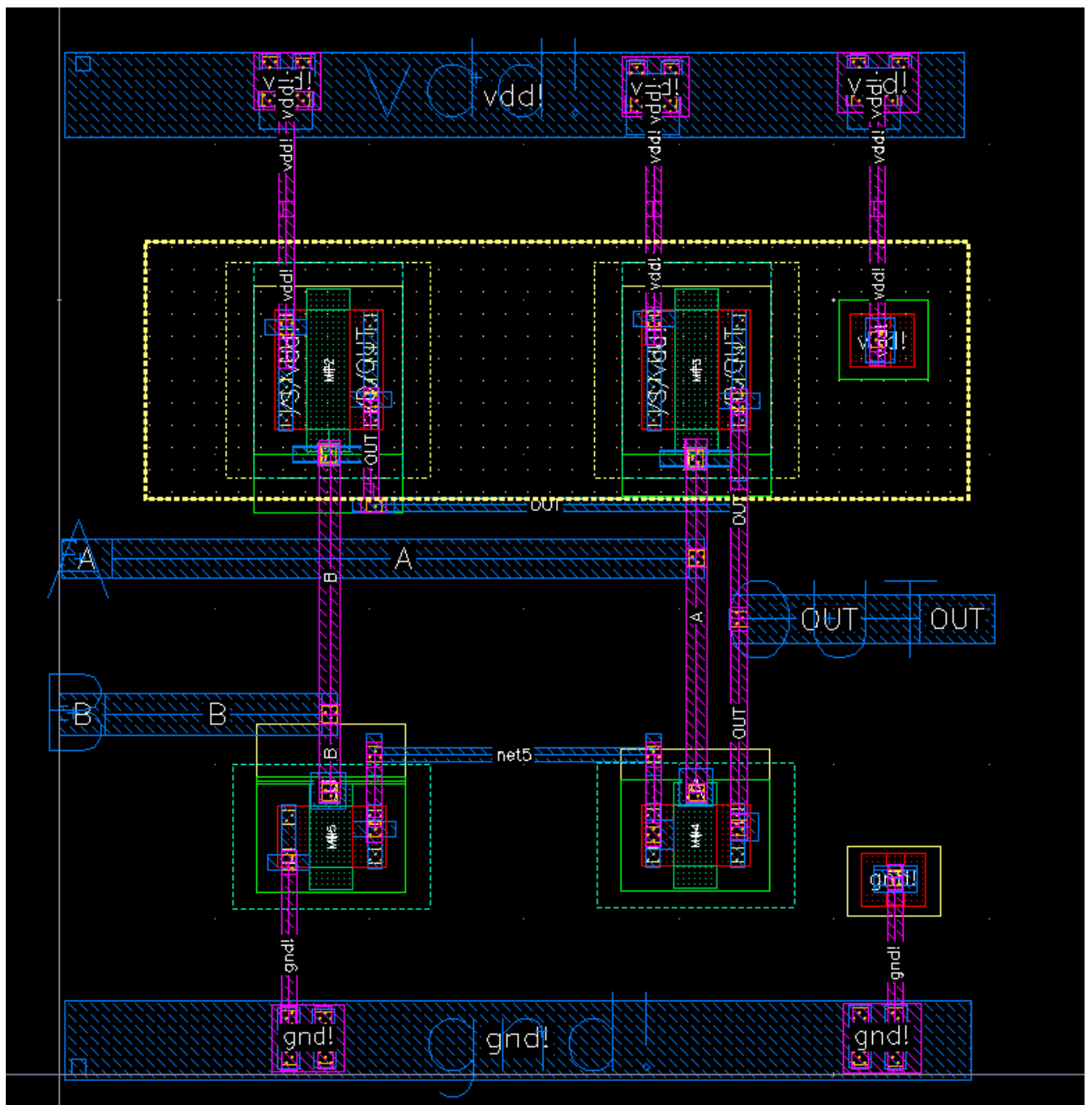
#####
#                                                                    #
# Run Result                : MATCH                                #
#                                                                    #
# Run Summary               : [INFO] ERC Results: Empty          #
#                           : [INFO] Extraction Clean            #
#                                                                    #
# ERC Summary File          : OR.sum                               #
# Extraction Report File    : OR.rep                              #
# Comparison Report File    : OR.rep.cls                          #
#                                                                    #
#####

```

Фиг. 3.15 Резултат от LVS на OR

3.4.3 Layout на NAND елемент

Физическата реализация на NAND елемента (фиг. 3.16) е проектирана с минимални размери съгласно технологичните правила. DRC проверките са проведени успешно, без установени нарушения (фиг. 3.17). LVS верификацията потвърждава пълно съответствие между схема и layout (фиг. 3.18).



Фиг. 3.16 Физическа реализация на NAND

```

Total CPU Time           : 4(s)
Total Real Time          : 4(s)
Peak Memory Used         : 44(M)
Total Original Geometry  : 154(286)
Total DRC RuleChecks     : 2260
Total DRC Results        : 0 (0)
Summary can be found in file NAND.sum
ASCII report database is /home/chavdara_beleva/Diplomna/diplomna/DRC/NAND/NAND.drc_errors.ascii
Checking in all SoftShare licenses.

Design Rule Check Finished Normally. Wed Jan 31 13:48:56 2018

```

Фиг. 3.17 Резултат от DRC на NAND

```

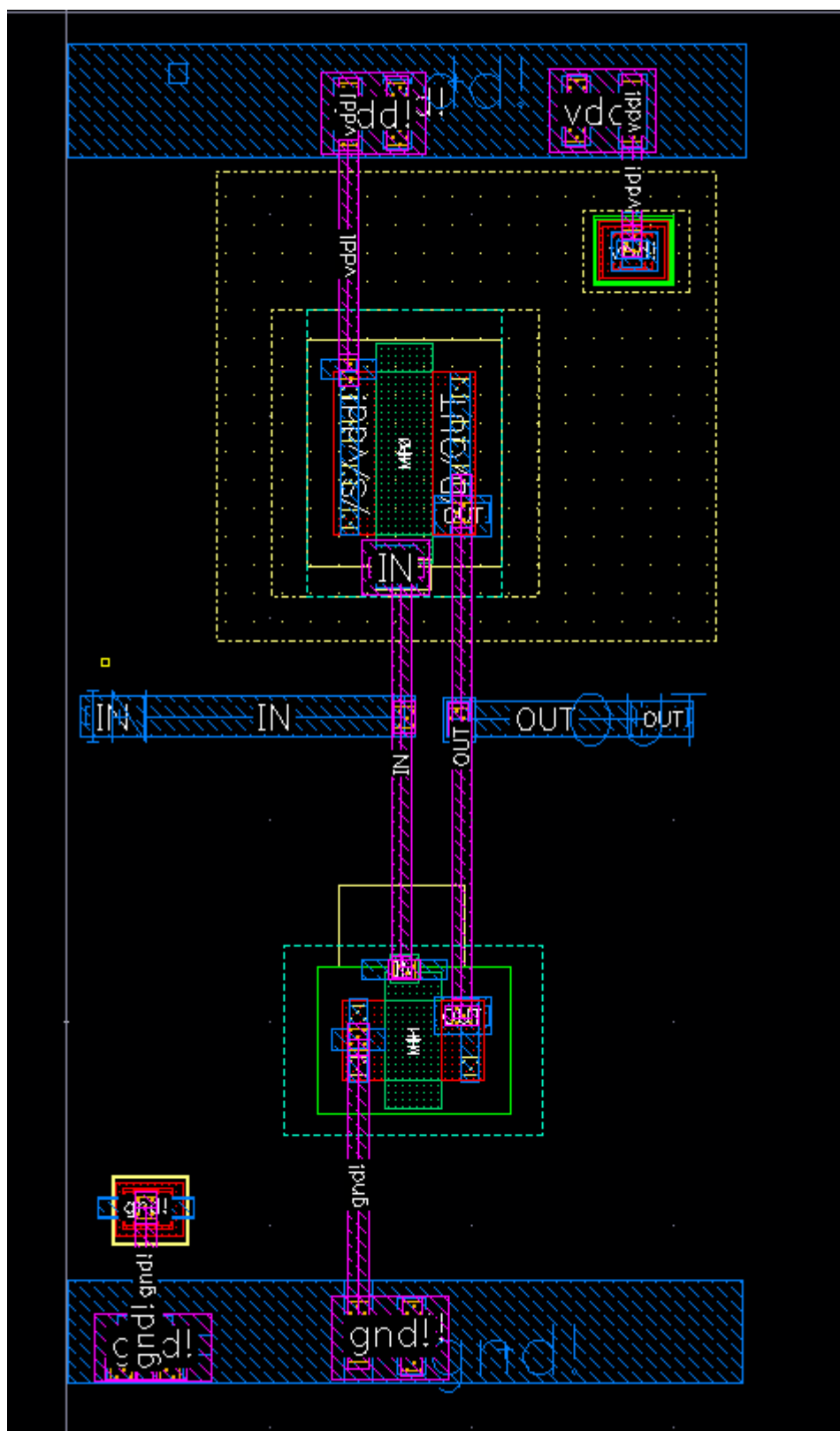
#####
#                                                                    #
# Run Result                : MATCH                                #
#                                                                    #
# Run Summary               : [INFO]  ERC Results: Empty          #
#                           : [INFO]  Extraction Clean             #
#                                                                    #
# ERC Summary File          : NAND.sum                             #
# Extraction Report File    : NAND.rep                             #
# Comparison Report File    : NAND.rep.cls                         #
#                                                                    #
#####

```

Фиг. 3.18 Резултат от LVS на NAND

3.4.4 Layout на NOT елемент

Физическата реализация на NAND елемента (фиг. 3.19) е проектирана с минимални размери съгласно технологичните правила. DRC проверките са проведени успешно, без установени нарушения (фиг. 3.20). LVS верификацията потвърждава пълно съответствие между схема и layout (фиг. 3.21).



Фиг. 3.19 Физическа реализация на NOT

```

Total CPU Time           : 4(s)
Total Real Time          : 4(s)
Peak Memory Used         : 44(M)
Total Original Geometry  : 134(165)
Total DRC RuleChecks     : 2260
Total DRC Results        : 0 (0)
Summary can be found in file NOT.sum
ASCII report database is /home/chavdara_beleva/Diplomna/diplomna/DRC/NOT/NOT.drc_errors.ascii
Checking in all SoftShare licenses.

```

Design Rule Check Finished Normally. Wed Jan 31 13:54:53 2018

Фиг. 3.20 Резултат от DRC на NOT

```

#####
#
# Run Result           : MATCH
#
# Run Summary          : [INFO] ERC Results: Empty
#                      : [INFO] Extraction Clean
#
# ERC Summary File     : NOT.sum
# Extraction Report File : NOT.rep
# Comparison Report File : NOT.rep.cls
#
#####

```

Фиг. 3.21 Резултат от LVS на NOT

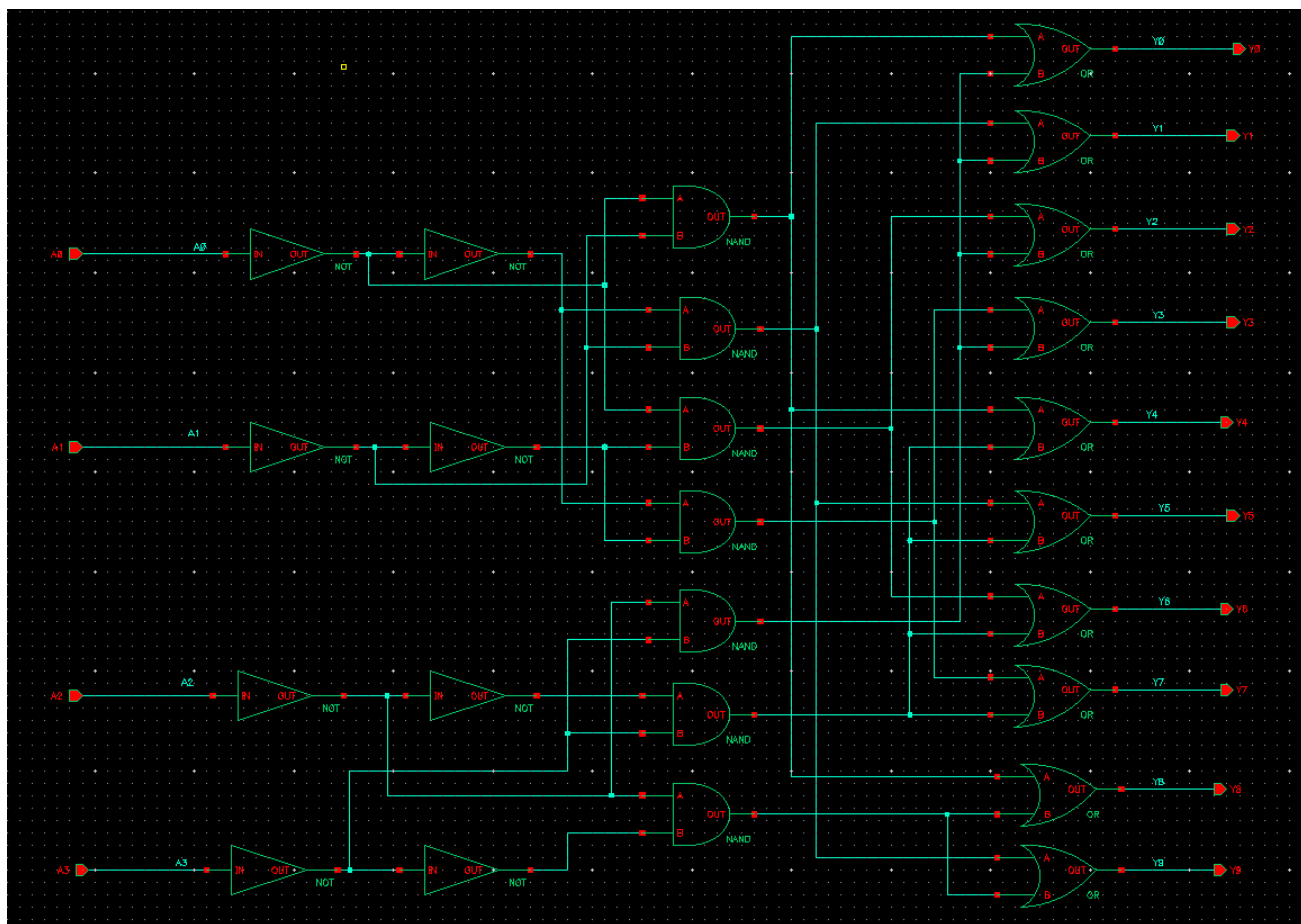
ЧЕТВЪРТА ГЛАВА

Схемотехнически и физически проект на 74НС42 и верификация на дизайна

4.1 Схема на 74НС42

4.1.1 Представяне на създадената схема

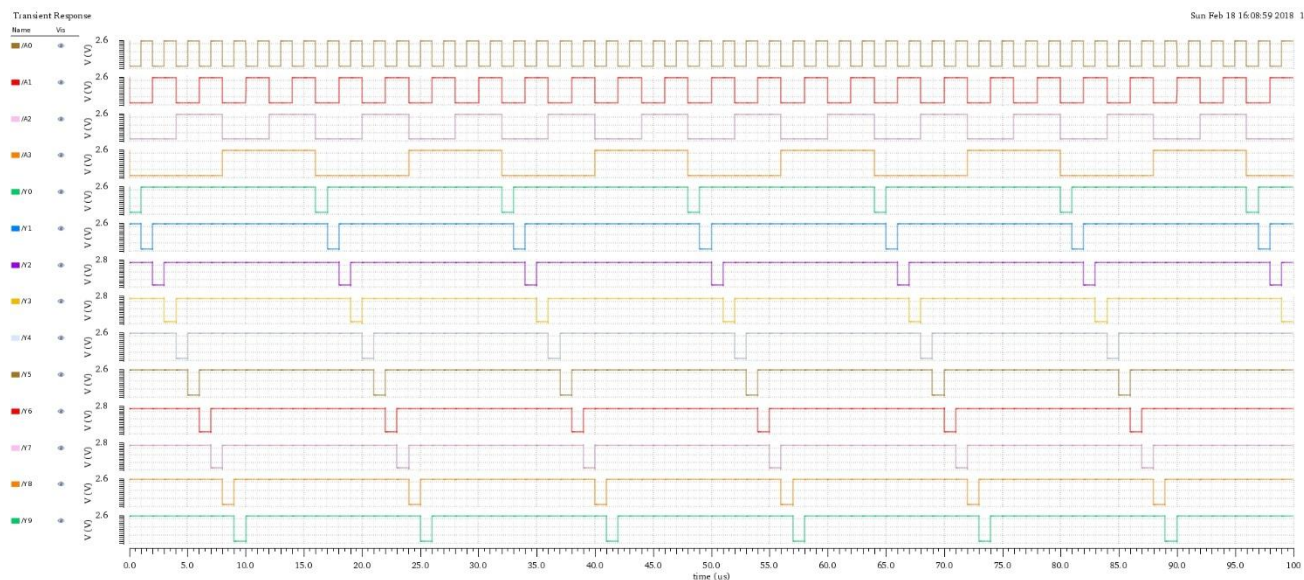
Проектираната схема на 74НС42 (*фиг. 4.1*) представлява сложна цифрова структура, реализирана като декодер с четири входни линии (А0-А3) и десет изходни линии (Y0-Y9). В основата на схематичното решение стои комбинация от логически елементи, започваща с NOT елементи при входните сигнали. Тези инвертори са разположени като двойки, като първият инвертор приема входния сигнал, а вторият осигурява допълнително буфериране и възстановява целостта на сигнала. В средната част на схемата са разположени NAND логически елементи, които изпълняват основната декодираща функция. Изходната секция е изградена от OR елементи, които формират финалните изходни сигнали. Цялостната архитектура на схемата е оптимизирана за минимално закъснение на сигналите и максимална надеждност на операциите, като същевременно е съобразена с изискванията за лесна реализация във физическия дизайн.



Фиг. 4.1 Схема на 74HC42

4.1.2 Резултати от симулацията на преходни процеси

Извършената симулация на преходните процеси демонстрира отлично функциониране на проектираната схема при всички възможни входни комбинации. Анализът на входно-изходните зависимости потвърждава точната работа на декодера, като за всяка входна комбинация се наблюдава активиране на точно един изход, докато останалите остават в неактивно състояние. Особено важно е да се отбележи, че всички преходни процеси се установяват в рамките на очакваните времеви интервали, което гарантира надеждна работа в реални условия. Симулацията обхваща пълния набор от входни комбинации, като потвърждава правилната функционалност на схемата при всички възможни работни режими.(фиг. 4.2)



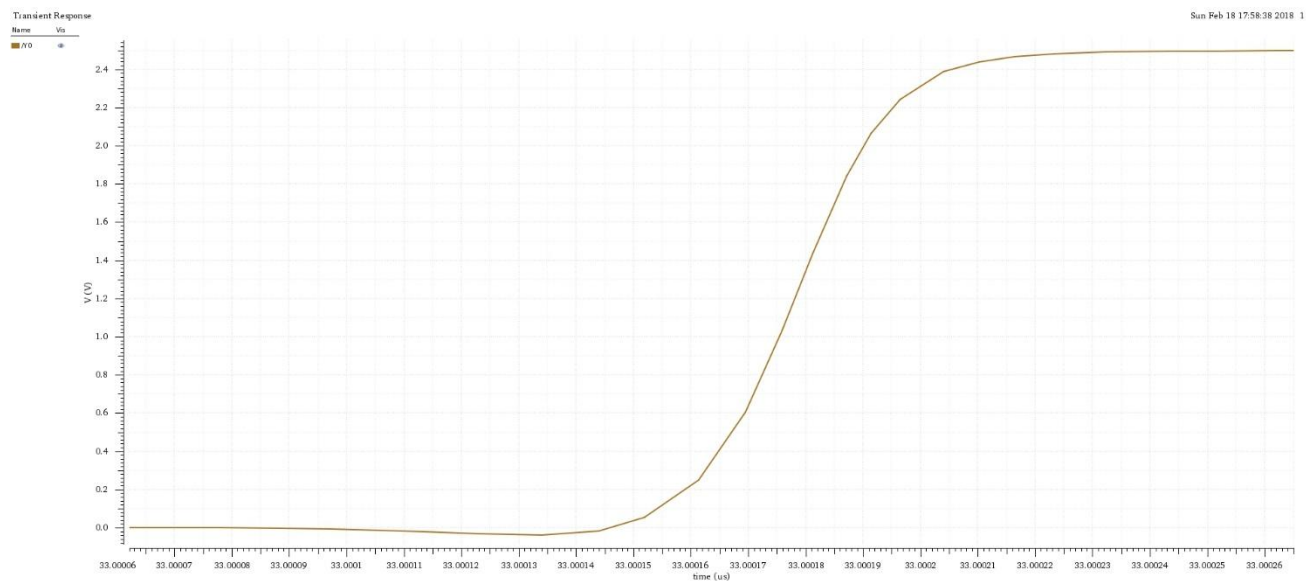
Фиг. 4.2 Резултати от симулацията на преходните процеси

4.1.3 Сравнителен анализ на резултатите преди и след PEX

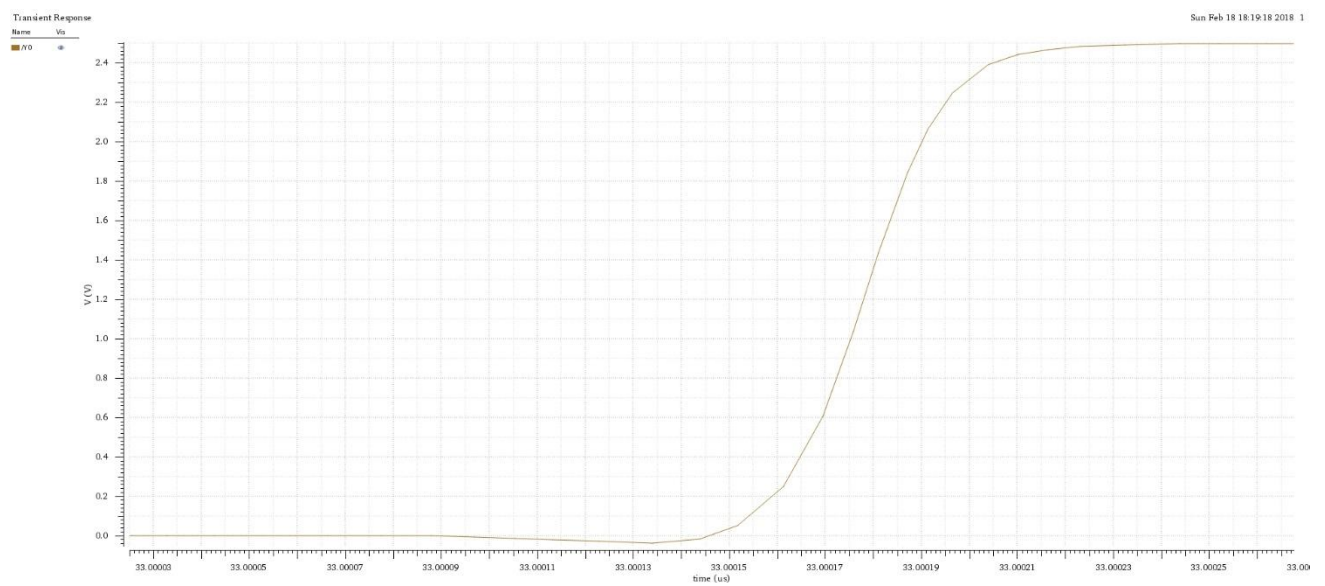
Извършеният сравнителен анализ на симулационните резултати преди (фиг. 4.3) и след (фиг. 4.4) PEX показва запазване на основната функционалност на схемата, като се наблюдават само минимални разлики в динамичното поведение на сигналите.

В първоначалната симулация сигналът демонстрира по-бърз преходен процес. След PEX анализа се наблюдава леко забавяне в преходния процес поради добавените паразитни капацитети и съпротивления, което се вижда в по-плавното нарастване на сигнала.

Важно е да се отбележи, че отклоненията са в приемливи граници и не нарушават работата на схемата. Финалната установена стойност на изходното напрежение остава практически непроменена, което показва, че паразитните ефекти влияят основно върху динамичните характеристики, а не върху статичното поведение на схемата.



Фиг. 4.3 Резултати от симулацията на преден фронт на сигнала преди PEX

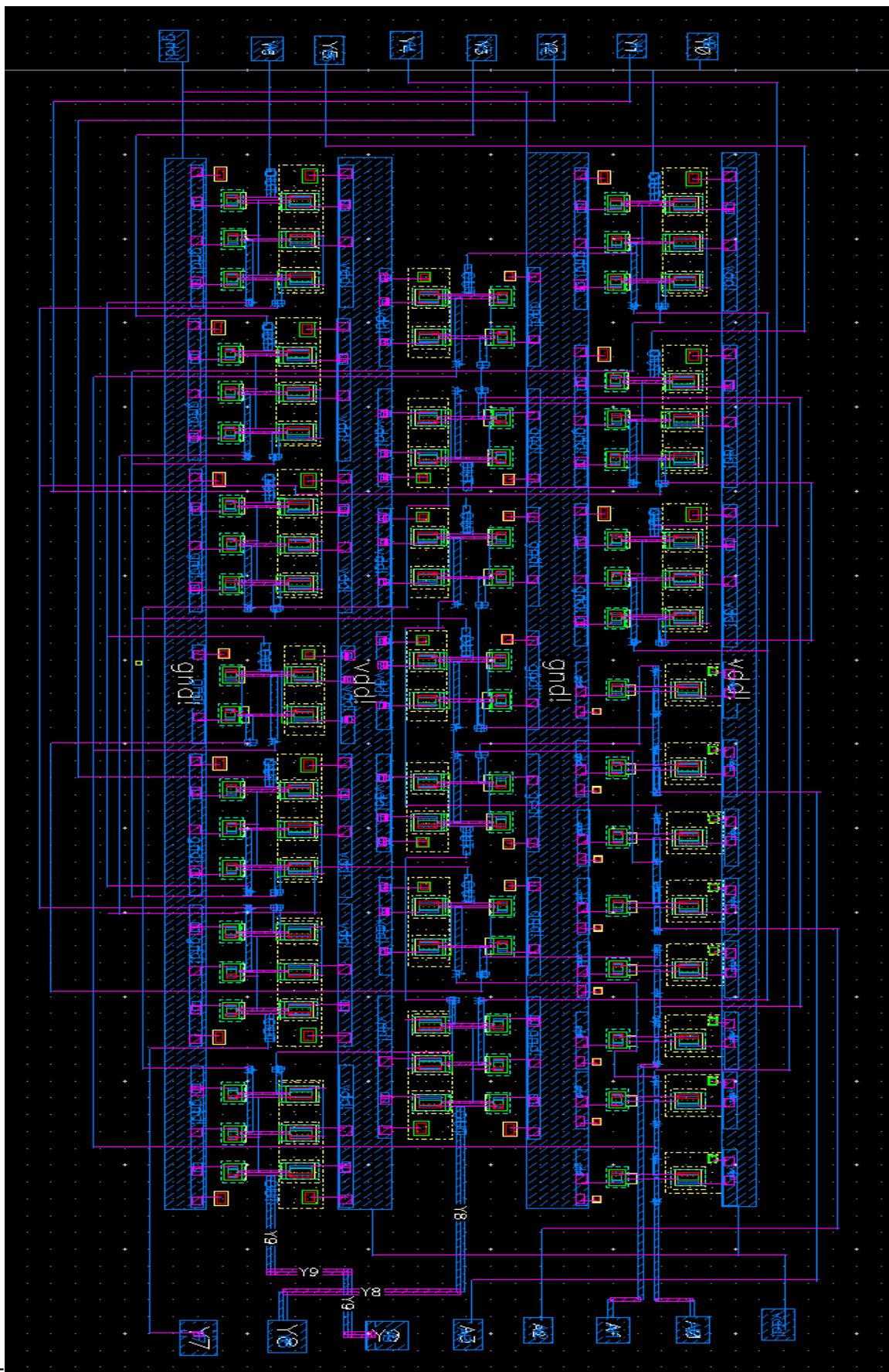


Фиг. 4.3 Резултати от симулацията на преден фронт на сигнала след PEX

4.2 Физическо проектиране (Layout)

4.2.1 Описание на създадения layout

Физическата реализация на 74НС42 е разработена с особено внимание към оптималното използване на пространството и ефективното разпределение на компонентите. Захранващите линии vdd! и gnd! са стратегически позиционирани, за да осигурят равномерно захранване на всички части от схемата. Транзисторите са разположени по начин, който минимизира общата площ на чипа, като същевременно улеснява свързването между отделните логически елементи. Трасирането на сигналните пътища е извършено с прецизност, като са избегнати ненужни кръстосвания и са оптимизирани дължините на връзките. Специално внимание е отделено на симетрията в дизайна, което не само подобрява производствените характеристики, но и допринася за по-добра работа на схемата. (*фиг. 4.3*)



Фиг. 4.3 Физическа реализация на 74НС42

4.2.2 Резултати от DRC проверката

Design Rule Check анализът на физическия дизайн показва отлични резултати, потвърждавайки пълното съответствие с технологичните изисквания за производство. Извършени са множество проверки, покриващи всички аспекти на дизайна - от минимални разстояния между елементите до специфични технологични ограничения. Всички проверки са преминати успешно, без установени нарушения на правилата за проектиране, което потвърждава качеството на изпълнението и производствената готовност на дизайна. (фиг. 4.4)

```
Total CPU Time           : 4(s)
Total Real Time          : 4(s)
Peak Memory Used         : 44(M)
Total Original Geometry  : 782(8067)
Total DRC RuleChecks     : 2260
Total DRC Results        : 0 (0)
Summary can be found in file 74HC42_3.sum
ASCII report database is /home/chavdara_beleva/Diplomna/diplomna/DRC/74HC42/74HC42_3.drc_errors.ascii
Checking in all SoftShare licenses.
```

Design Rule Check Finished Normally. Wed Jan 31 19:39:49 2018

Фиг. 4.4 DRC резултат на 74HC42

4.2.3 Резултати от LVS проверката

Сравнението между схемата и физическия дизайн (Layout vs. Schematic) демонстрира пълно съответствие, отразено в резултат "MATCH". Този резултат потвърждава, че всички компоненти от схемата са коректно имплементирани във физическия дизайн, като връзките между тях са реализирани точно според схематичното представяне. Процесът на екстракция премина без проблеми, показвайки чист резултат без наличие на грешки или несъответствия. Цялостната йерархична структура на дизайна е запазена непроменена при прехода от схема към физическа

реализация, което е изключително важно за правилното функциониране на крайното устройство. (фиг. 4.5)

```
#####  
#                                                                 #  
# Run Result              : MATCH                                #  
#                                                                 #  
# Run Summary             : [INFO]  ERC Results: Empty         #  
#                         : [INFO]  Extraction Clean            #  
#                                                                 #  
# ERC Summary File        : 74HC42_3.sum                        #  
# Extraction Report File  : 74HC42_3.rep                        #  
# Comparison Report File  : 74HC42_3.rep.cls                    #  
#                                                                 #  
#####
```

Фиг. 4.5 LVS резултат на 74HC42

ЗАКЛЮЧЕНИЕ

В резултат на проведената изследователска и практическа работа в рамките на настоящата дипломна работа е осъществена цялостна разработка и реализация на декодер 74НС42 в интегрална форма, като са изпълнени всички поставени цели и технически изисквания съгласно първоначалното задание.

Теоретичното изследване обхваща систематичен анализ на съществуващите декодерни архитектури, техните функционални характеристики и области на приложение. Извършеното проучване на BCD-към-десетичните декодери и спецификите на серията 74НС предоставя необходимата теоретична база за последващата практическа реализация на проекта.

В практическата част са разработени фундаменталните логически елементи OR, NAND и NOT, които представляват основните градивни компоненти на декодера. Посредством прецизен DC и времеви анализ е потвърдена коректността на тяхното функциониране. Оптимизацията на размерите на транзисторите е осъществена с цел постигане на оптимално съотношение между производителност и енергийна ефективност.

Физическата реализация на логическите компоненти е извършена при строго съблюдаване на технологичните изисквания, което е верифицирано чрез успешното провеждане на DRC и LVS проверки. Особено внимание е отделено на оптималното позициониране на елементите и ефективното трасиране на междусъединенията.

Заклучителният етап от разработката - имплементацията на пълния декодер 74НС42, демонстрира успешната интеграция на всички проектирани компоненти в единна функционална структура. Симулационните анализи на преходните процеси потвърждават коректната функционалност на декодера при всички входни комбинации, а физическият дизайн удовлетворява всички верификационни изисквания.

СЪДЪРЖАНИЕ

УВОД.....	3
Използвани съкращения.....	5
ПЪРВА ГЛАВА.....	7
Теоретично проучване.....	7
1.1 Декодер – аналози и решения	7
1.1.1 Видове декодери.....	7
1.1.1.1 Two-to-four Line Decoder (Двоично-към-четворично декодиране)	7
1.1.1.2 Three-to-eight Line Decoder (Троично-към-осмично декодиране)	8
1.1.1.3 Four-to-sixteen Line Decoder (Четирибитов към 16-линиен декодер) ..	9
1.1.1.4 BCD-to-Decimal Decoder (BCD-към-десетичен декодер).....	11
1.1.1.5 BCD-to-7-Segment Decoder (BCD-към-7-сегментен декодер)	12
1.1.1.6 Address Decoder (Адресен декодер).....	14
1.1.2 Приложения на декодерите в цифровата електроника	14
1.1.2.1 Приложение в компютърни системи и памети.....	14
1.1.2.2 Приложение в системи за виртуализация	15
1.1.2.3 Приложение в измервателната техника.....	15
1.1.2.4 Приложение в индустриални системи.....	16
1.1.2.5 Приложение в интерфейсни системи.....	16
1.1.2.6 Приложение в навигационни системи	16
1.1.2.7 Приложение в роботизирани системи.....	17
1.1.3 Сравнение между различните видове декодери	17
1.1.3.1 Сравнение спрямо входно-изходното пространство.....	17
1.1.3.2 Сравнение спрямо технологията на изработка	17
1.1.3.3 Сравнение спрямо специализацията	18
1.1.3.4 Сравнение спрямо възможностите за разширяемост.....	18
1.1.3.5 Сравнение спрямо управляващите възможности	18
1.1.4 Интерфейс със други цифрови компоненти.....	19
1.1.4.1 Начини на свързване с други цифрови схеми	19
1.1.4.2 Входно/изходни нива на сигналите и съвместимост.....	19
1.1.4.3 Управляващи сигнали	19
1.1.4.4 Каскадно свързване на декодери	20
1.1.4.5 Интерфейс с различни видове дисплеи и индикатори	21
1.1.4.6 Буфериране на входове/изходи	22
1.1.4.7 Защита от шум и смущения	23
1.1.5 Анализ на съществуващи решения и архитектури:.....	25
1.1.5.1 Преглед на популярни интегрални схеми с декодери (74xx серия) ..	25

1.1.5.1.1	74HC138/74HCT138	25
1.1.5.1.2	74HC42/74HCT42	25
1.1.5.1.3	74HC154/74HCT154	25
1.1.5.2	Сравнение на различни архитектури	26
1.1.5.3	Анализ на предимства и недостатъци на всяко решение.....	27
1.1.5.4	Методи за оптимизация.....	28
1.1.5.5	Съвременни тенденции в дизайна на декодери.....	29
1.1.5.6	Критерии за избор на конкретно решение според приложението.....	31
1.2.	Цифрови логически компоненти	32
1.2.1	Основни логически елементи (AND, OR, NOT, NAND, NOR, XOR)	32
1.2.1.1	AND	32
1.2.1.2	OR	33
1.2.1.3	NOT	34
1.2.1.4	NAND	35
1.2.1.5	NOR.....	35
1.2.1.6	XOR.....	36
1.2.1.7	XNOR	37
1.2.2	Комбинационни схеми	38
1.2.2.1	Логически примитиви.....	38
1.2.2.2	Мултиплексори и демултиплексори	38
1.2.2.3	Декодери и енкодери	39
1.2.2.4	Суматори	40
1.2.2.5	Компаратори	41
1.2.3	Последователни схеми.....	42
1.2.3.1	Тригери	42
1.2.3.2	Регистри	45
1.2.3.3	Броячи	46
1.2.3.4	Памети	47
1.2.4	Анализ на входно-изходни характеристики.....	49
1.2.4.1	Логически нива.....	49
1.2.4.2	Шум и допустими отклонения	50
1.2.4.3	Хистерезис	50
1.2.4.4	Входни и изходни импеданси.....	50
1.2.4.5	Fan-in/Fan-out параметри	51
1.2.5	Времеви параметри и закъснения	51
1.2.5.1	Време за разпространение на сигнала	51
1.2.5.2	Време за нарастване и спадане	52
1.2.5.3	Време на установяване и време на задържане	52
1.2.5.4	Максимална работна честота	52

1.2.5.5	Времеви диаграми и анализ	53
1.2.6	Натоварвателна способност	53
1.2.6.1	Максимален изходен ток	53
1.2.6.2	Брой входове, които могат да се управляват от един изход	54
1.2.6.3	Изходно съпротивление	54
1.2.6.4	Необходимост от буфериране	54
1.2.6.5	Влияние на капацитивен товар.....	55
1.2.6.6	Разсейвана мощност при различни натоварвания	55
1.3	Проектиране на интегрални схеми	55
1.3.1	Методология за проектиране	55
1.3.2	Йерархични нива на проектиране	56
1.3.3	Етапи в проектирането на интегрални схеми.....	57
1.3.4	Верификация и тестване	58
1.3.5	Оптимизация на дизайна	59
1.4	Работа със системата за автоматизирано проектиране Cadence Virtuoso	60
1.4.1	Въведение в Cadence Virtuoso	60
1.4.2	Основни инструменти и функционалност	61
1.4.2.1	Schematic Editor	61
1.4.2.2	Layout Editor	61
1.4.2.3	ADE	62
1.4.3	Създаване на схеми.....	62
1.4.4	Симулация и анализ	63
1.4.5	Проектиране на layout.....	64
1.4.6	DRC	64
1.4.7	LVS	65
1.4.8	PEX.....	66
1.5	Process Design Kit	67
1.5.1	Структура на PDK.....	67
1.5.2	Компоненти и модели	68
1.5.3	Технологични библиотеки.....	69
1.5.4	Design rules	70
1.5.5	Екстракция на паразити	71
1.5.6	Интеграция с Cadence среда.....	72
ВТОРА ГЛАВА		74
Принцип на работа на 74НС42		74
ТРЕТА ГЛАВА.....		77
Практическа реализация и анализ на логически елементи.....		77
3.1	Проектиране на основни логически елементи.....	77
3.1.1	Реализация на OR елемент.....	77

3.1.1.1 Схемно представяне	77
3.1.1.2 Описание на функционалността	78
3.1.2 Реализация на NAND елемент	79
3.1.2.1 Схемно представяне	79
3.1.2.2. Описание на функционалността	80
3.1.3 Реализация на NOT елемент	80
3.1.3.1 Схемно представяне	80
3.1.3.2 Описание на функционалността	81
3.2 DC анализ и оразмеряване на логическите елементи	81
3.2.1 Методология за DC анализ	81
3.2.2 DC анализ и оразмеряване на OR елемент	82
3.2.3 DC анализ и оразмеряване на NAND елемент	83
3.2.4 DC анализ и оразмеряване на NOT елемент	84
3.3 Времеви (Transient) анализ на логическите елементи	85
3.3.1 Методология за времеви анализ	85
3.3.2 Времеви анализ на OR елемент	86
3.3.3 Времеви анализ на NAND елемент	87
3.3.4 Времеви анализ на NOT елемент	88
3.4 Проектиране на физическа реализация (Layout)	88
3.4.1 Методология за проектиране на layout	88
3.4.2 Layout на OR елемент	90
3.4.3 Layout на NAND елемент	91
3.4.4 Layout на NOT елемент	93
ЧЕТВЪРТА ГЛАВА	96
Схемотехнически и физически проект на 74НС42 и верификация на дизайна ...	96
4.1 Схема на 74НС42	96
4.1.1 Представяне на създадената схема	96
4.1.2 Резултати от симулацията на преходни процеси	97
4.1.3 Сравнителен анализ на резултатите преди и след PEX	98
4.2 Физическо проектиране (Layout)	100
4.2.1 Описание на създадения layout	100
4.2.2 Резултати от DRC проверката	102
4.2.3 Резултати от LVS проверката	102
ЗАКЛЮЧЕНИЕ	104
СЪДЪРЖАНИЕ	105
Използвана литература	109

Исползвана литература

1. Baker, R. J. (2019). CMOS Circuit Design
2. Brunvand, E. (2010). Digital VLSI Chip Design with Cadence and Synopsys CAD Tools. Addison-Wesley.
3. Floyd, T. L. (2019). Digital Fundamentals
4. Harris, D. M., & Harris, S. L. (2022). Digital Design and Computer Architecture
5. Plummer, J. D., Deal, M. D., & Griffin, P. B. (2000). Silicon VLSI Technology: Fundamentals, Practice, and Modeling. Prentice Hall
6. Rabaey, J. M., Chandrakasan, A., & Nikolić, B. (2016). Digital Integrated Circuits: A Design Perspective (3rd ed.). Pearson
7. Sedra, A. S., & Smith, K. C. (2014). Microelectronic Circuits (7th ed.). Oxford University Press
8. Sze, S. M., & Ng, K. K. (2006). Physics of Semiconductor Devices (3rd ed.). Wiley-Interscience
9. Tocci, R. J., Widmer, N. S., & Moss, G. L. (2023). Digital Systems
10. Wakerly, J. F. (2017). Digital Design: Principles and Practices
11. Weste, N. H. E., & Harris, D. (2016). CMOS VLSI Design: A Circuits and Systems Perspective (4th ed.). Pearson
12. Wolf, S. (2004). Microchip Manufacturing. Lattice Press
13. IEEE Standard 1364-2005: Verilog Hardware Description Language
14. IEEE Standard 1800-2017: SystemVerilog
15. Cadence Design Systems (2023). Virtuoso Platform User Guide
16. NXP Semiconductors (2023). 74HC/HCT/HCU Logic Family Specifications
17. ON Semiconductor. CMOS Logic Family Documentation
18. LOGIC Pocket Data Book (Rev. B)
19. Texas Instruments (2023). 74HC/HCT Logic Family Data Sheet Collection
20. CD54HC154, CD74HC154, CD54HCT154, CD74HCT154 datasheet (Rev. D)
21. Texas Instruments (2023). 74HC42 Technical Documentation

22. 74HC4511; 74HCT4511 BCD to 7-segment latch_decoder_driver
23. Bushnell, M., & Agrawal, V. (2022). Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits. Springer.
24. [Pedroni, V. A. \(2020\). Digital Electronics and Design with VHDL. Morgan Kaufmann](#)
25. [Zwolinski, M. \(2019\). Digital System Design with SystemVerilog. Pearson](#)
26. [Ashenden, P. J. \(2008\). Digital Design: An Embedded Systems Approach Using VHDL. Morgan Kaufmann](#)
27. [EDA for IC Implementation, Circuit Design, and Process Technology](#)
28. [Raghunathan, A., Jha, N. K., & Dey, S. \(2018\). High-Level Power Analysis and Optimization. Springer](#)