

Praktikum Schaltungsintegration (PSI)

Versuch Nr.: SI1
Versuchstitel: Full-Custom-Entwurf von CMOS-Zellen
Versuchskonzept: C. Schimpfle / D. Kohlert

I. Versuchsziel

In den Versuchen SI1 bis SI3 sollen Sie den Umgang mit einem sehr leistungsfähigen, in der Industrie ein-gesetzten EDA (Electronic Design Automation)-Software-Paket kennen lernen. Sie werden den Einsatz der Programme anhand von vollständigen Entwurfsabläufen (Design Flows) für Full-Custom- und Standardzell-Entwürfe durchführen.

II. Versuchskurzbeschreibung

Im ersten Versuch werden Sie sich mit der Bedienung des Programms für die Layouterstellung (CADENCE Virtuoso) vertraut machen. Sie sollen einen CMOS-Inverter mit minimalem Flächenbedarf unter Beachtung aller geometrischen Entwurfsregeln entwerfen.

Im Lauf der Durchführung des Versuchs (Kap. V.3) werden unterschiedliche Themen behandelt:

- Bedienung von Virtuoso
 - Einführung in die Bedienung des Layoutprogramms VIRTUOSO XL
- Layout von n- und p-Kanal-Transistor mit SKILL
 - Hier werden die grundlegenden Rechtecke der Transistoren dimensioniert und platziert. Hierfür ist vor allem ein gründliches Verständnis der Design Rules nötig. Die Platzierung der Rechtecke erfolgt über Skripten
- Kontaktlöcher
 - Die Kontaktlöcher werden als Zeitgründen als fertige Zellen vorgegeben, die benötigten Fähigkeiten würden V.3.2 entsprechen
- Design der Stromschienen für VDD und VSS
 - Hier wird der Aufbau der Versorgungsschienen mit den Substratkontakten behandelt. Diese werden auf der Basis der vorgegebenen Kontaktlöcher im klassischen "Hand"-Layout erstellt. Somit werden auch diese Funktionen der Software kennengelernt
- Aufbau des Inverters
 - Hier wird die Platzierung und Verdrahtung mehrerer Basiszellen zu einer kompletten Standardzelle durchgeführt.
- Aufbau des Ringoszillators
 - Hier wird die Platzierung und Verdrahtung mehrerer Inverter zu einem 5-stufigen Ringoszillator durchgeführt

III. Versuchsvorbereitung

1. Machen Sie sich mit dem verwendeten Prozess vertraut ! Vollziehen Sie alle aufgeführten Prozessschritte nach !
2. Studieren Sie die Entwurfsregeln (Design Rules). Tragen Sie im Protokollfile (psi1_prot.ods) die Angaben zu V.3.2 ein und beantworten Sie die Fragen zu V.3.3 und V.3.4 !
3. Informieren Sie sich über (statische) CMOS-Logik (z.B. in /1/) !

IV. Entwurststile

Beim Entwurf einer integrierten Schaltung sind eine Reihe von Anforderungen zu erfüllen: Die korrekte Funktion der Schaltung muss über alle Entwurfsteilschritte gewährleistet sein, die durch die Technologie gegebenen Randbedingungen müssen berücksichtigt werden und die Möglichkeiten der Technologie sollen möglichst gut genutzt werden. Andererseits soll der für den Entwurf aufzubringende Zeit- und Personalaufwand möglichst klein sein. All diese Anforderungen sind gleichzeitig nur schwer einzuhalten. Es wurden daher Methoden entwickelt, die abhängig von der jeweiligen Entwurfsaufgabe möglichst optimale Kompromisslösungen anbieten.

Man unterscheidet zwischen drei unterschiedlichen Entwurststilen:

- dem **voll-kundenspezifischen Entwurf (full-custom design)**,
- dem **zellorientierten Entwurf** und
 - dem **Array-Entwurf**.
 -

Der **voll-kundenspezifische Entwurf** ist der Entwurfstil mit den meisten Freiheitsgraden. Jeder Transistor kann einzeln dimensioniert und platziert werden. Die die Transistoren verbindenden Leiterbahnen können bezüglich elektrischer Randbedingungen (z.B. Signallaufzeiten, Übersprechen usw.) optimal ausgelegt werden. Die charakteristischen Eigenschaften dieses Entwurfstils sind:

- hohes Optimierungspotenzial,
- optimale Technologieausnutzung,
- hohes Fehlerrisiko,
- hoher Erstellungs- und Korrekturaufwand,
- geringer Automatisierungsgrad,
- lange „time to market“.

(Die „time to market“ ist die Zeitdauer, die zwischen Abgabe der Kundenspezifikationen und Auslieferung des ersten spielfähigen Schaltungsmusters vergeht.) Die aufgezählten Eigenschaften zeigen insgesamt, dass dieser Entwurfstil nur dann sinnvoll ist, wenn die damit erstellte Schaltung über einen längeren Zeit-raum in hohen Stückzahlen verkauft werden kann. Auch für das Erstellen von Zellbibliotheken und den Entwurf von Speicherzellen, also von Strukturen, die vielfach in anderen Schaltungen verwendet werden, rentiert sich der hohe Erstellungsaufwand.

Beim **zellorientierten Entwurf** wird auf bereits vorentworfene Grundlayouts (Bibliothekszellen) zurückgegriffen. Diese Zellen sind hinsichtlich der Korrektheit ihres Entwurfs verifiziert und ihr elektrisches Verhalten ist bekannt. Während es beim so genannten **Makrozellentwurf** i.d.R. keine Einschränkungen bezüglich der Geometrie des Zelllayouts gibt, ist das Zelllayout beim so genannten **Standardzellentwurf** erheblichen Einschränkungen unterworfen. Gewöhnlich werden rechteckige Zellen identischer Höhe verwendet, die am oberen und unteren Zellenrand über Versorgungsleitungen verfügen. Durch einfaches Aneinanderreihen (**Abutment**) dieser Standardzellen können komplexere Funktionsblöcke realisiert werden. Die Spannungsversorgung der Zellreihen erfolgt kammartig von der linken und der rechten Seite. Abb. 1 zeigt ein einfaches Beispiel eines Standardzellentwurfs.

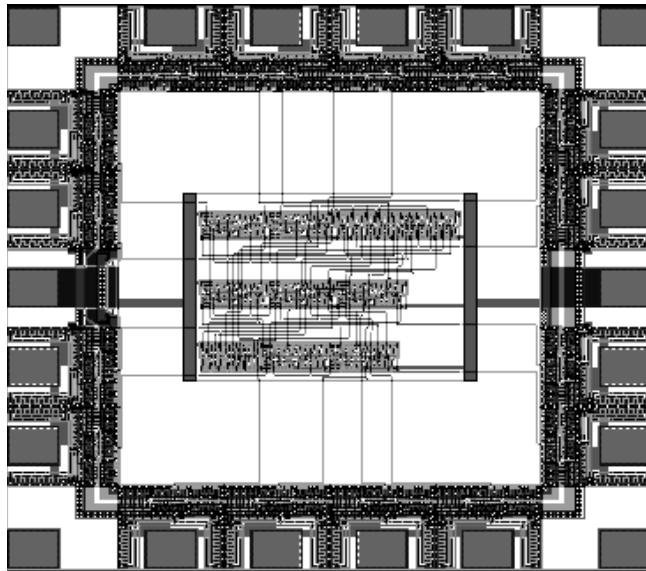


Abb. 1: Einfacher Standardzellentwurf (L-Edit-Demobeispiel)

Zwischen den einzelnen Zellreihen gibt es Verdrahtungskanäle, in denen die Verdrahtung der Zellen untereinander vorgenommen wird (**Channel Routing**).

Die charakteristischen Eigenschaften des Standardzellentwurfs sind:

- einfache Zellanordnung,
- einfache Verdrahtung,
- hoher Automatisierungsgrad,
- wenig Optimierungsmöglichkeiten,
- schlechte Ausnutzung der Chipfläche,
- separate Behandlung des Chiprandes notwendig,
- kürzere „time to market“ als beim voll-kundenspezifischen Entwurf.

Der Standardzellentwurf wird für den Entwurf digitaler Schaltungen heute sehr häufig eingesetzt.

Um die „time to market“ noch weiter zu verkürzen, wurde nach Entwurfsstilen gesucht, die eine Vorfertigung von Siliziumscheiben erlauben. Dieser Weg wird beim so genannten **Gate-Array-Entwurf** beschritten. Bei diesem Entwurfsstil werden Scheiben so weit vorgefertigt, dass bereits alle Transistoren vorhanden sind. Die „Personalisierung“ des Chips erfolgt dadurch, dass die Verdrahtung der Transistoren kundenspezifisch vorgenommen wird. Der Hauptvorteil dieses Entwurfsstils gegenüber den bisher genannten ist die Verkürzung der Fertigungszeit aufgrund der Vorfertigung. Von den drei genannten Entwurfsstilen hat dieser i.d.R. die kürzeste „time to market“.

V. Versuchsdurchführung

V.1 Einloggen am PC

V.2 Aufruf von Virtuoso

V.3 Layout des CMOS-Inverters:

Aufgabenstellung

Ihre Aufgabe ist es, das Layout eines statischen CMOS-Inverters zu erstellen. Dieses Layout soll alle geometrischen Entwurfsregeln entsprechend der Vorlesung SI erfüllen und dabei so wenig Fläche wie möglich beanspruchen.

V.3.1 Bedienung von Virtuoso

Die Bedienung von Virtuoso ist in der Virtuoso-Kurzanleitung (VirtuosoTut.pdf) beschrieben.

V.3.2 Layout von n- und p-Kanal-Transistor mit SKILL

Der verwendete Prozess (Skywater 130nm) weist Design-Rules auf, die ein Handlayout äusserst mühsam machen. Es ist daher sinnvoller, sich anhand einer Skizze und den relevanten Design-Rules Größe und Position der benötigten Rechtecke zu überlegen und diese dann mit einem SKILL-Skript automatisiert zu erstellen. Dies gilt vor allem für die beiden Transistoren. Wenn die Transistoren gezeichnet sind, ist die Situation einfacher, da man sich an den Positionen der Transistoren orientieren kann.

"SKILL" ist eine Cadence-eigene Skriptsprache mit allen Features einer Programmiersprache und der Möglichkeit, Zeichenobjekte zu erstellen. In unserem Fall werden nur die Befehle zum erstellen und platzieren von Rechtecken und einige einfache Berechnungen benötigt.

Beispiel: SKILL-Skript zum Erstellen eines Rechtecks in der Ebene "diff" "drawing"

Die ersten beiden Zeilen bleiben immer gleich:

```
cur_view = geGetEditCellView( hiGetCurrentWindow( ) )
println( cur_view )
```

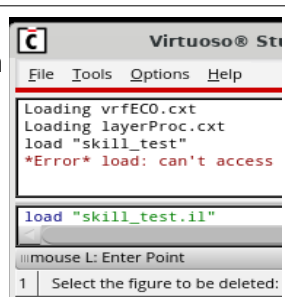
Für das Zeichnen angegeben werden müssen die linke untere und die rechte obere Ecke sowie die Ebene. Der Zeichenbefehl lautet:

```
dbCreateRect(cur_view '("diff" "drawing") list(xmin : ymin xmax : ymax)
```

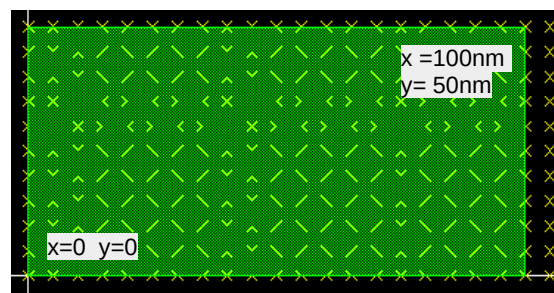
Das folgende Skript zeichnet ein Rechteck in die Drawing-Ebene "Diffusion" mit den Eckkoordinaten 0µm ; 0µm (unten links) und 50nm ; 100nm (oben rechts):

```
cur_view = geGetEditCellView( hiGetCurrentWindow( ) )
println( cur_view )
; Rechteck Diffusion
xmin = 0
ymin = 0
xmax = 0.1
ymax = 0.05
dbCreateRect(cur_view '("diff" "drawing") list(xmin : ymin xmax : ymax)
)
```

Das SKILL-Skript muss im Arbeitsverzeichnis von Virtuoso stehen. Die nebenstehende Abb. zeigt den Aufruf:



Erzeugt wird folgendes Rechteck (diff = grün):



Die Möglichkeit, Koordinaten zu berechnen, kann für viele Zwecke gewinnbringend eingesetzt werden.

Es gibt z. B. eine Design-Regel (n_psd.5a), nach der die Ebene nsdm die Ebene diff auf jeder Seite um 125nm überlappen muss. Das folgende Skript zeigt die Erstellung des Rechtecks in der Ebene nsdm aufgrund der Koordinaten des Rechtecks in diff :

```
cur_view = geGetEditCellView( hiGetCurrentWindow( ) )
println( cur_view )

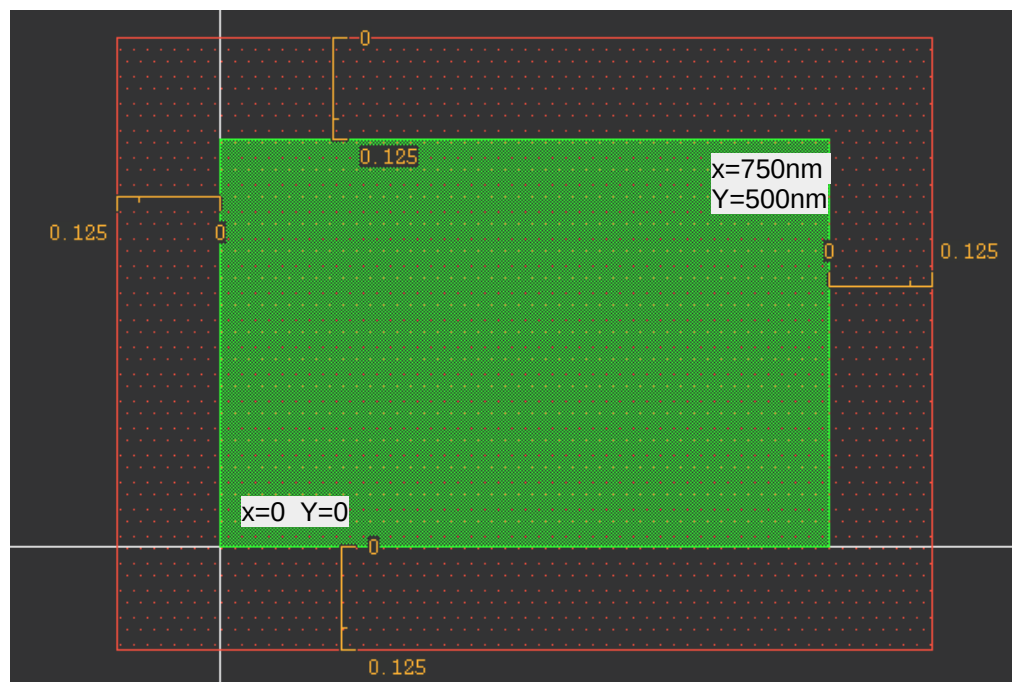
n_psd_5a = 0.125

;Rechteck diff
xmin = 0
ymin = 0
xmax = 0.75
ymax = 0.5
dbCreateRect( cur_view '("diff" "drawing") list(xmin : ymin xmax : ymax) )

;n_psd
xmin = -n_psd_5a
ymin = -n_psd_5a
xmax = 0.75 + n_psd_5a
ymax = 0.5 + n_psd_5a
dbCreateRect( cur_view '("nsdm" "drawing") list(xmin : ymin xmax : ymax) )
```

Der Name der Design Rule für die Überlappung kann als Variablenname verwendet werden, damit ergeben sich flexible Skripte, die man sogar einfach an geänderte Design Rules anpassen könnte.

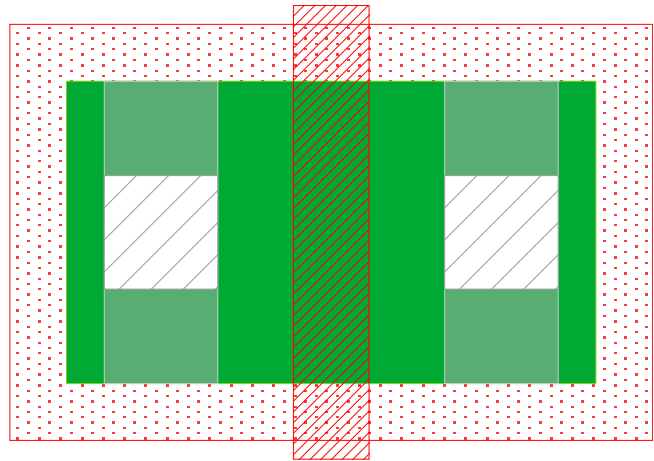
Resultat:



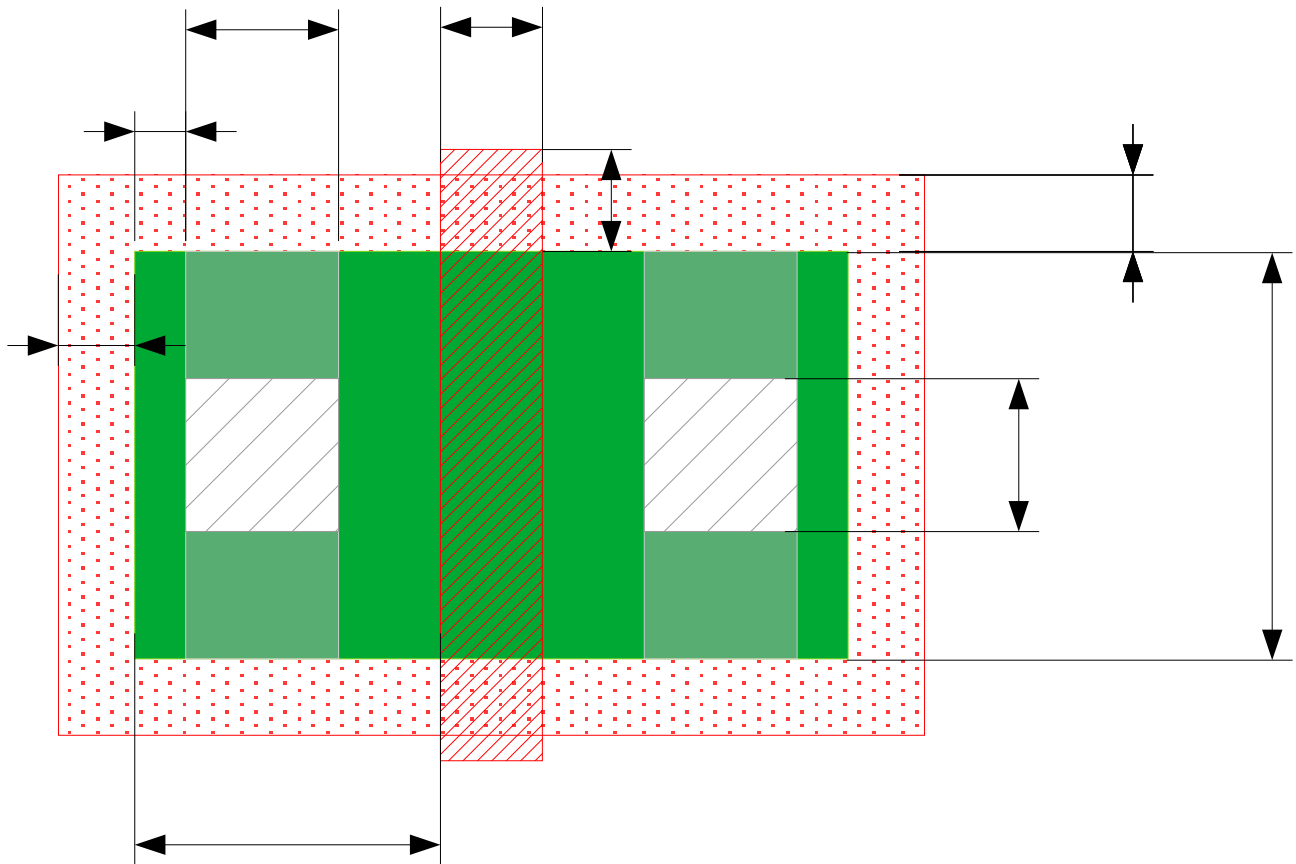
n-Kanal-Feldeffekttransistor

Die folgende Abb. zeigt den Aufbau eines n-Kanal-Feldeffekttransistors:

Beschriften Sie im Protokollfile alle Rechtecke mit den Namen des zugehörigen Layer !



Tragen Sie im Protokollfile für jede der in der folgenden Zeichnung eingezeichneten Maßlinien den Namen der zugehörigen Design Rule und den Zahlenwert ein !



Es ist ein SKILL-File (nfet_psi.il) vorgegeben. Das File ist ein einfaches Textfile und kann mit einem Texteditor bearbeitet werden.

Bis auf die Ebenen diff und nsdm sind alle Rechtecke mit den nötigen Angaben vorhanden.

Die Abmessungen für die Ebenen diff und nsdm müssen noch eingetragen werden. Ergänzen Sie die Ebenen um die nötigen Angaben !

Benutzen Sie den Namen der jeweiligen Design Rule als Variablennamen, so dass in den Zeichen-Anweisungen für die Rechtecke nur noch Variablennamen stehen !

Die linke untere Ecke von diff soll der Nullpunkt sein.

Öffnen Sie in Virtuoso eine neue Zelle mit dem Namen nfet_psi (Layout) und führen Sie das Skript aus !

Testen Sie Ihr Skript und führen Sie einen Design Rule Check durch ! Falls Fehlermeldungen auftreten, muss das File solange korrigiert werden, bis keine Fehlermeldungen mehr auftreten !

Beachten Sie beim Austesten der Zellen, dass Sie die jeweils die vorhandene Zelle im Layout löschen müssen, bevor Sie das Skript ausführen.

Speichern Sie Ihre Zelle ab.

Kopieren Sie Ihre Ergänzungen des Skripts und das Layout in das Protokollfile psi1_prot.ods. Die Erstellung eines .png-Files ist im Tutorial beschrieben.

Skript Testen

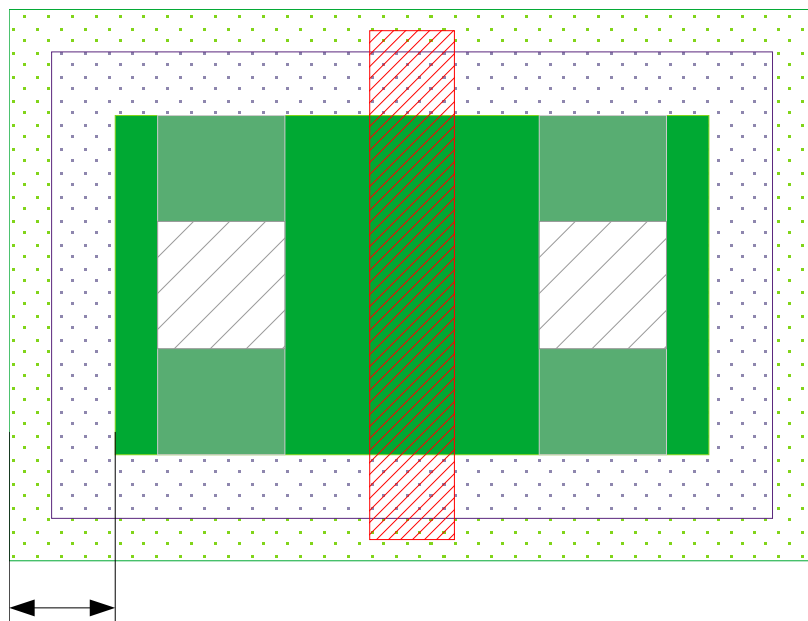
Design Rule Check

Falls Fehlermeldungen auftreten, muss das File solange korrigiert werden, bis keine Fehlermeldungen mehr auftreten

Beachten Sie Beim Austesten der Zellen, dass Sie jeweils die vorhandene Zelle im Layout löschen müssen, bevor Sie das Skript ausführen.

p-Kanal-Feldeffekttransistor

Die folgende Abb. zeigt den Aufbau eines p-Kanal-Feldeffekttransistors:



Beschriften Sie im Protokollfile alle Rechtecke mit den Namen des zugehörigen Layer !

Geben Sie für die eingezeichnete Maßlinie im Protokollfile den Namen der zuständigen Design

Rule und den Wert an !

Das SKILL-File (pfet_psi.il) ist ebenfalls vorgegeben.

Auch hier müssen die Ebenen diff und psdm ergänzt werden.

Testen Sie Ihr Skript und führen Sie einen Design Rule Check durch ! Falls Fehlermeldungen auftreten, muss das File solange korrigiert werden, bis keine Fehlermeldungen mehr auftreten !

p-Kanal-Transistoren sind in der Regel bei gleichen Abmessungen hochohmiger als n-FETs.

Es ist daher sinnvoll, bei den p-FETs eine größere Kanalweite zu wählen.

Welche Variable muss verdoppelt werden, damit ein p-FET mit doppelter Kanalweite entsteht ?

Führen Sie die Modifikation durch, und überprüfen Sie das Resultat mit dem DRC !

Speichern Sie Ihre Zelle ab !

Kopieren Sie das Skript und das Layout in das Protokollfile psi1_prot.ods

V.3.3 Kontaktlöcher

Um die Transistoren kontaktieren zu können, werden Kontaktlöcher benötigt, die ihre eigenen Design Rules aufweisen.

Frage V3.3.1:

Mit welcher Metall-Lage können Diffusionsgebiete und Polysiliziumleitungen kontaktiert werden ?

Frage V3.3.2:

Welche Ebenen benötigt der Kontakt zwischen der untersten Metall-Ebene und einem Diffusionsgebiet ?

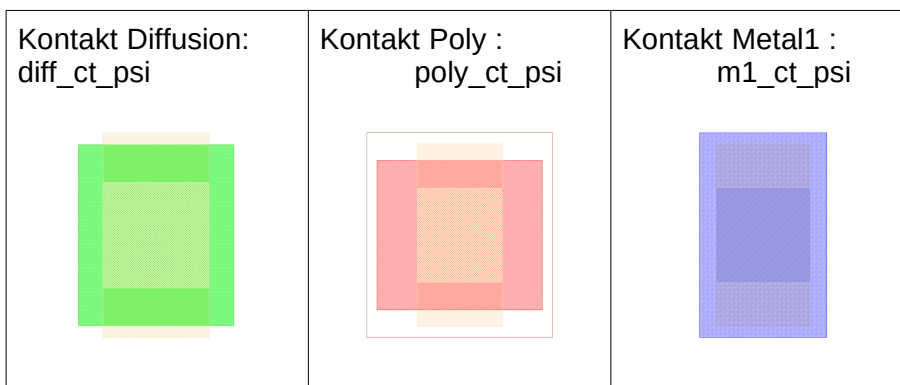
Frage V3.3.3:

Welche Ebenen benötigt der Kontakt zwischen der untersten Metall-Ebene und einer Polysiliziumleitung ?

Frage V3.3.4:

Welche Ebenen benötigt der Kontakt zwischen der untersten Metall-Ebene und einer Metall-1-Leitung ?

Aus Zeitgründen werden die Zellen für die Kontakte zur Verfügung gestellt:



Es stehen jetzt alle Zellen zur Verfügung, um einen Inverter layouten zu können.
Es fehlen aber noch die "Stromschienen" zur Spannungsversorgung des Inverters und die Substratkontakte.

V.3.4 Design der Stromschienen für VDD und VSS

TAP-Kontakt p-Substrat

Frage V.3.4.1:

Welche Ebenen werden benötigt, um einen Substratkontakt für das p-Substrat zu erstellen ?
Welche der beiden Versorgungsspannungen muss angeschlossen werden ?

Erstellen Sie eine leere Zelle mit dem Namen p_tapcon_psi !
Zunächst muss eine Verbindung zwischen li1 und TAP erzeugt werden. Diese kann aus der Zelle diff_ct_psi erstellt werden:

- Zelle diff_ct_psi in Layout laden
- Zelle drehen, damit die li1-Bahn horizontal wird
- Zelle diff_ct_psi "flach machen (displayed levels)", um die einzelnen Rechtecke editieren zu können
- Rechteck vom Layer diff in Layer tap transferieren.

Die Breite der Stromschiene soll in etwa der der FET's entsprechen. Mit dieser Vorgabe können 2 Kontakte nebeneinander in die Zelle platziert werden.

Frage V.3.4.2:

Welche Design Rule legt den Mindestabstand von licon fest, wie groß ist er ?
Welche Design Rule legt den Mindestabstand von mcon fest, wie groß ist er ?

Kopieren die bis jetzt erstellte Struktur und platzieren Sie sie neben die vorhandene.
Bewegen Sie die zweite Zelle so, dass der Abstand dem Mindestabstand entspricht !
Der nötige Abstand lässt sich vorher mit dem Ruler einzeichnen.
Schliessen Sie die Lücke in den Ebenen (am einfachsten mit "Stretch") !
Führen Sie einen DRC durch, beseitigen Sie evtl. Fehler !

Frage V.3.4.3:

Die Ebene "psdm" muss oberhalb der Ebene "tap" liegen. Welche Design Rule gibt den Wert der Überlappung an, wie groß ist er ?

Markieren Sie die entsprechenden Abstände mit Rulers und zeichnen Sie das Rechteck ein:

- Rechteck um "tap" herum einzeichnen, das ungefähr passt
- Exakte Justierung an den Rulers mit "Stretch"

Führen Sie einen DRC durch, beseitigen Sie evtl. Fehler !

Die bisherige Zelle kontaktiert die "tap"-Ebene mit "li1". Zur großflächigen Verdrahtung muss aber in Anschluss an "metal1" möglich sein.

Platzieren Sie jeweils einen passenden Kontakt (siehe Vorgabe) über den "tap"-li1-Kontakt.
Stacked Contacts sind erlaubt.

Führen Sie einen DRC durch, beseitigen Sie evtl. Fehler !

Kopieren Sie das Layout mit Hilfe eines Screenshots in das Protokollfile !

TAP-Kontakt n-Well

Speichern Sie die Zelle "p_tapcon_psi" unter dem Namen "n_tapcon_psi" ab. Verschieben sie das Rechteck in der Ebene "psdm" in die Ebene "nsdm" !

Bauen Sie die n-Well entsprechend der obigen Antwort ein und führen Sie einen DRC durch !

Frage V.3.4.4:

Welche Design Rule ist jetzt für die Platzierung der n-Well relevant ? Wert ?

Es tritt eine Fehlermeldung auf. Korrigieren Sie die Abmessungen der p-Well entsprechend der Fehlermeldung !

Führen Sie einen DRC durch und speichern Sie die Zelle ab !

Kopieren Sie das Layout mit Hilfe eines Screenshots in das Protokollfile !

V.3.5 Aufbau des Inverters

Erstellen Sie eine neue Zelle (Name inv_psi).

Platzieren Sie die bisher erstellten Zellen, so dass sich ein Inverter mit minimalen Abmessungen realisieren lässt ! Beachten Sie, dass in Standardzellen alle Ein- und Ausgänge auf die Metal1-Ebene herausgeführt werden müssen ! Die Spannungsversorgung erfolgt über die oben entwickelten Stromschienen.

Verdrahten Sie die Zellen so, dass sich eine funktionsfähige Schaltung ergibt. Die Verdrahtung kann sowohl mit Rechtecken als auch mit der "wire"-Funktion durchgeführt werden.

Führen Sie einen DRC durch und speichern Sie die Zelle ab !

Kopieren Sie das Layout in das Protokollfile !

V.3.6 Aufbau des Ringoszillators

Erstellen Sie eine neue Zelle (Name ringo_psi).

Platzieren Sie die bisher erstellten Zellen, so dass sich ein 5-stufiger Ringoszillator mit minimalen Abmessungen realisieren lässt ! Die Spannungsversorgung erfolgt über die oben entwickelten Stromschienen.

Verdrahten Sie die Zellen so, dass sich eine funktionsfähige Schaltung ergibt. Die Verdrahtung kann sowohl mit Rechtecken als auch mit der "wire"-Funktion durchgeführt werden.

Führen Sie einen DRC durch und speichern Sie die Zelle ab !

Kopieren Sie das Layout in das Protokollfile !