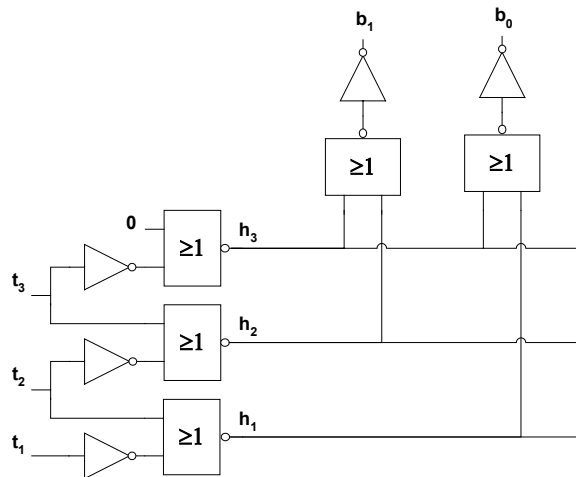


5. Digitale code omzetter (29 punten)

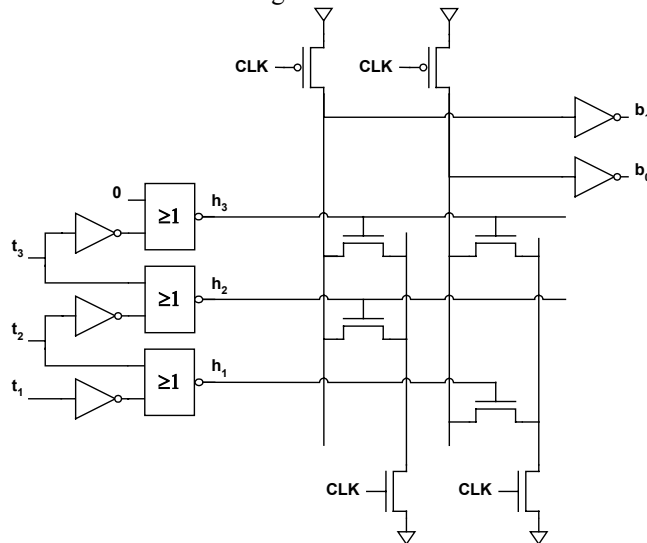
Voor deze vraag moeten de technologie gegevens van de draft 2^e editie van Rabaey gebruikt worden, zoals ook (gedeeltelijk) samengevat op de design-data bladen. De gehanteerde ontwerpregels komen uit de 1^e editie. λ is $0.125 \mu\text{m}$ en V_{DD} is 2.5 V .

Deze vraag zal de encoder (code omzetter) uit figuur 4 behandelen. Neem in het vervolg aan dat de besproken AD-omzetter een 6-bits omzetter is. De encoder logica krijgt als ingangen de $2^6 - 1 = 63$ uitgangen van de verzadigende operationele versterkers en produceert een 6-bits binaire codering van het sensor signaal. Merk op dat de uitgangen van de operationele versterkers een zogenaamde thermometer code afgeven: als bit i hoog is zijn alle "lagere" bits ook hoog. De encoder heet dan ook wel een *thermometer-naar-binair* omzetter. Zo'n omzetter kan elegant gemaakt worden door de thermometer code eerst om te zetten naar een 1-hoog code, waarbij het aantal bits gelijk blijft maar alleen het hoogste aan-bit van de thermometer code aan gezet wordt in de 1-hoog code. Alle andere bits zijn dan laag. Deze 1-hoog code stuurt dan weer een aantal NOR poorten (één per uitgangsbite) die de inverse van de uitgangsbite produceren, 6 invertoren leveren uiteindelijk het binaire signaal $\langle b_5-b_0 \rangle$. Iedere NOR-poort heeft 32 ingangen, omdat van de (6-bits) binaire getallen van 0 t/m 63 er voor iedere bit-positie er steeds 32 met een "0" en 32 met een "1" zijn.

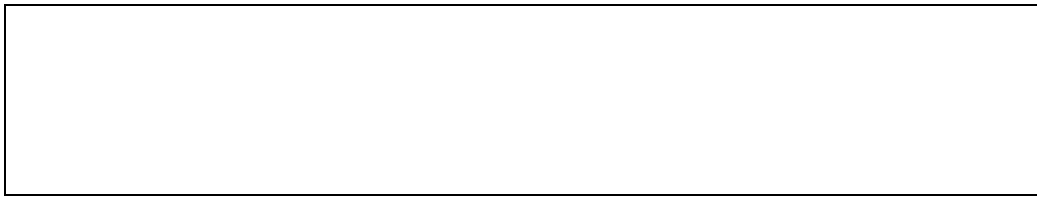
Het principe is hieronder geïllustreerd voor een 2-bits AD omzetter. Hier vormt $\langle t_3-t_1 \rangle$ de thermometer code, $\langle h_3-h_1 \rangle$ de 1-hoog code en $\langle b_1-b_0 \rangle$ het binaire signaal.



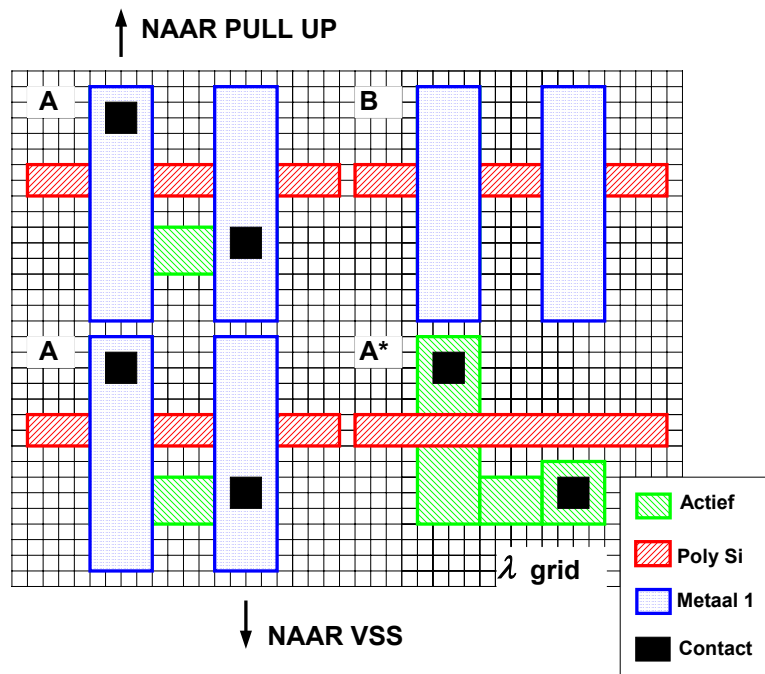
Voor de NOR-poorten die gestuurd worden door $\langle h_3-h_1 \rangle$ zullen we dynamische CMOS poorten gebruiken. Het schema voor 2 bits wordt dan als volgt.



5.1 Waarom verdient deze vorm van logica in dit geval de voorkeur boven statische CMOS logica?



De layout van de NOR-poorten heeft een regelmatige opbouw, het pull-down gedeelte bestaat uit 2 verschillende typen “tegels” zoals gegeven in onderstaande figuur. Tegel *A* is met pull-down transistor, en tegel *B* is zonder pull-down transistor. In de figuur is tegel *A** hetzelfde als tegel *A*, maar het metaal masker is niet getekend om de precieze vorm van het actieve gebied te kunnen bepalen. Het grid is een λ -grid. Natuurlijk horen in de uiteindelijke layout de tegels tegen elkaar geschoven te worden. Iedere rij correspondeert met een bepaalde uitgangswaarde en iedere kolom met een uitgangsbij.



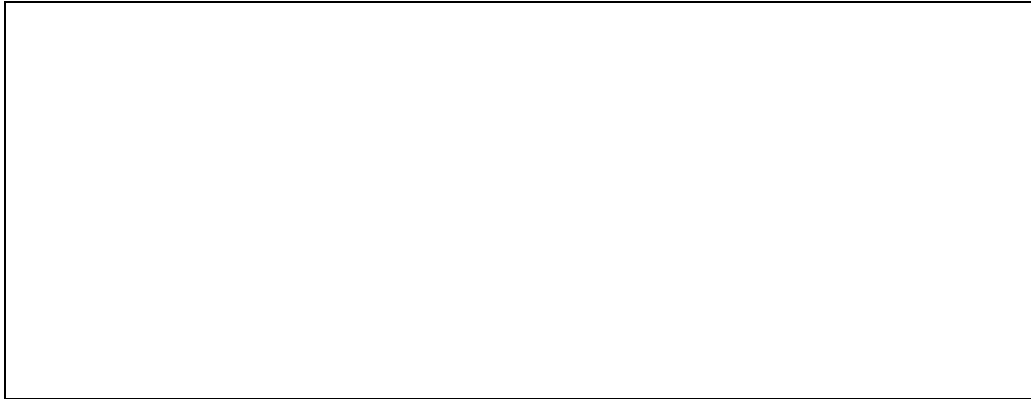
5.2 Bereken de effectieve capaciteit C_{eff} van tegel type *A* voor de evaluatietijd t_{eval} . Neem de juiste K_{eq} uit voorbeeld 5.3 van Rabaey draft 2^e editie. Neem alleen de bijdragen van onderstaande tabel en verwaarloos de capaciteit van het metaal, deze is klein t.o.v. de andere capaciteiten. Vul uw (deel-) uitkomsten in in onderstaande tabel.

Bijdrage	Afmetingen in λ, λ^2	Afmetingen in μ, μ^2	K_{eq}	C [fF]
Diffusie oppervlak	λ^2	μ^2		
Diffusie rand	λ	μ		
Gate overlap	λ	μ	1	
Totaal	XXX	XXX	XXX	

5.5 Teken een RC schema voor de worst-case precharge tijd, en een tweede schema voor de worst-case evaluatie tijd. Geef duidelijk de relevante parameters (component waarden) aan, kies daarvoor uit de onderstaande tabel. Andere capaciteiten en weerstanden (zoals

bijvoorbeeld de ingangscapaciteiten van de invertoren en bedradingsweerstand) mogen verwaarloosd worden. **N.B.** Met worst-case wordt in dit geval de tijd bedoeld die hoort bij de 1-hoog code waarvoor de vertraging het grootst is.

R_p	Aan-weerstand van precharge transistor
R_e	Aan-weerstand van evaluatie transistor
R_{pdn}	Aan-weerstand van pull-down transistor
C_A	Capaciteit van tegel type A
C_B	Capaciteit van tegel type B

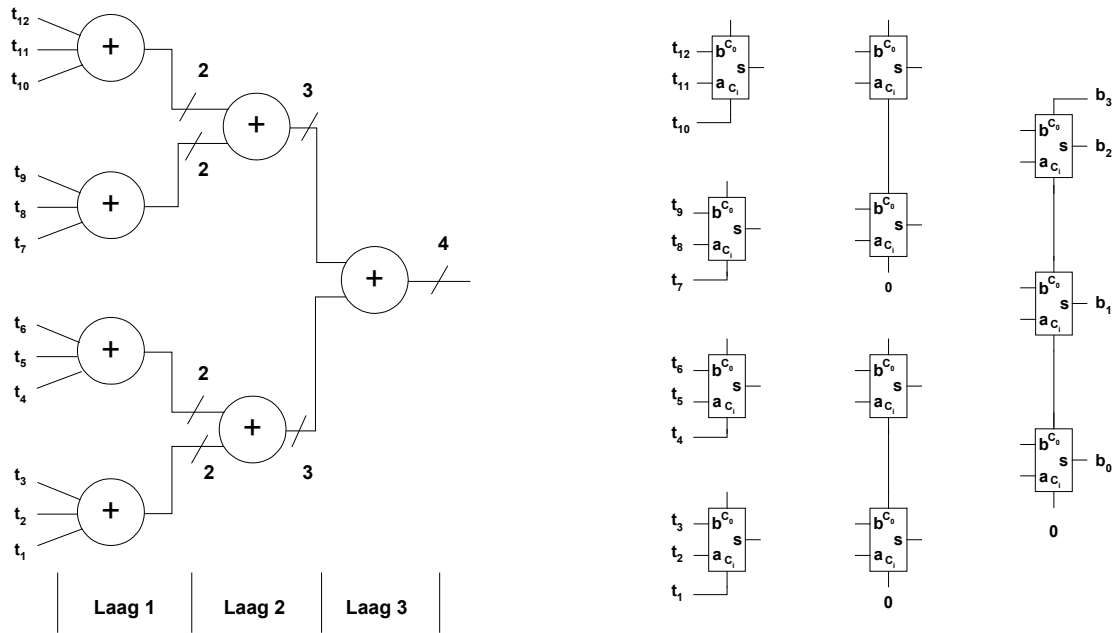


5.6 Stel nu dat de effectieve capaciteit van tegel A en B voor de laag-naar-hoog transitie gegeven worden door respectievelijk 3 fF en 0.5 fF. Dimensioneer de precharge transistor (geef W/L, bij minimum L) voor een worst-case precharge tijd van 0.75 ns.

W/L =
Berekening:

Een potentieel probleem bij flash AD omzeters zoals hier besproken is het optreden van zogenaamde luchtbel-fouten in de thermometer code. Bijvoorbeeld tengevolge van het niet gelijk zijn van alle vertragingen in de schakeling, zullen niet alle thermometer uitgangen precies op tijd schakelen. Het resultaat is dat er soms een nul kan staan tussen een reeks éénen. Dit lijkt op een luchtbel in een kwikbuis, vandaar de naam. Een luchtbelfout geeft natuurlijk aanleiding tot een foute waarde aan de binaire uitgang, omdat er meer dan één rij van de NOR poorten actief is, waarbij de fout relatief groot kan zijn. Ter voorkoming van dit probleem kan ook een andere manier van thermometer-naar-binair omzetten gekozen worden, nl. het tellen van het aantal enen. De fout die veroorzaakt wordt door een luchtbel is dan slechts gelijk aan één discretisatie niveau.

Het tellen van het aantal enen kan geschieden door ieder bit van de thermometer code te beschouwen als een 1-bit getal, en al die getallen bij elkaar op te tellen. Voor de optellers kun je dan normale full-adder cellen gebruiken, en om 63 bits bij elkaar op te tellen kun je de optelling onderverdelen in lagen. Iedere laag heeft een aantal optellers, maar dit aantal neemt af bij hogere lagen, terwijl de bitbreedte van iedere opteller toeneemt. In de tekening hieronder links is dit principe weergegeven en zijn de bitbreedtes aangegeven door de schuine streepjes met het cijfer.



5.7 Geef in de tekening hierboven rechts aan hoe de verschillende Full-Adder cellen met elkaar verbonden moeten worden, m.a.w. maak de schakeling af door de bedrading te tekenen.

5.8 De schakeling hierboven is, zoals getekend, geschikt voor een 12-bits thermometer code. Hoe kun je de schakeling modifieren, zonder extra componenten (zoals Full-Adders of andere logische poorten) te gebruiken, zodanig dat hij een 15-bits thermometer code kan omzetten naar binair?