

```
<a name="home">
</a>
<br />
<div style="text-align: center;">
<span style="font-family: Georgia, Times New Roman, serif;"><a name="Awal"> </a>
<a href="#Akhir">[MENUJU AKHIR]</a></span></div>
<span style="font-family: Georgia, Times New Roman, serif;"><span style="font-family:
'georgia'&quot; , 'times new roman'&quot; , serif;"><br /></span>
</span><br />
<center>
<div style="background-color: white; border: 2px dashed rgb(23, 128, 221); height: 240px; overflow:
auto; padding: 10px; text-align: center; width: 330px;">
<span style="font-family: Georgia, Times New Roman, serif;"><b>DAFTAR ISI</b>
</span><br />
<div style="text-align: left;">
<a href="#Tujuan">1. Tujuan</a></div>
<div style="text-align: left;">
<a href="#Alat">2. Alat dan Bahan</a></div>
<div style="text-align: left;">
<a href="#Dasar">3. Dasar Teori</a></div>
<div style="text-align: left;">
<a href="#Percobaan">4. Percobaan</a><br />
<div style="text-align: left;">
<a href="#Video">5. Video</a></div>
<div style="text-align: left;">
<a href="#Link">6. Link Download</a></div>
<div style="text-align: left;">
<div style="text-align: left;">
<div style="text-align: left;">
</div></div></div></div></div></center>
```

<div style="text-align: justify;">

<u></u>

1. Tujuan [Kembali]</div><ol style="text-align: left;">Mampu membuat aplikasi carry propagation Memahami fungsi komponen pada carry propagation

<div style="text-align: justify;">

<u></u>

2. Alat dan Bahan [Kembali]</div>

- Alat

Logicstate
<p style="text-align: center;">

</p><div class="separator" style="clear: both;"><div class="separator" style="clear: both; color: black;"><div style="margin-left: 1em; margin-right: 1em; text-align: center;"></div><br style="color: black; text-align: left;" /><div style="color: black; text-align: justify;">
</div><div style="color: black; text-align:

g/s222/XOR.jpg" style="margin-left: 1em; margin-right: 1em;"></div></div>
<div style="text-align: justify;">
</div><div style="text-align: justify;">Gerbang XOR ini memerlukan dua input untuk menghasilkan satu output. Jika input berlogika 1 jumlahnya ganjil maka output akan berlogika 1. Begitu juga sebaliknya, jika input berlogika 1 jumlahnya genap maka output akan berlogika 0. </div></div><p style="text-align: center;"></p><p></p>Gerbang AND<div class="separator" style="clear: both;"><div class="separator" style="clear: both;"><div style="margin-left: 1em; margin-right: 1em; text-align: center;"></div></div>
<div style="text-align: justify;">
</div><div style="text-align: justify;">Gerbang AND ini memerlukan dua atau lebih input untuk menghasilkan satu output. Jika semua atau salah satu inputnya merupakan bilangan biner 0, maka outputnya akan menjadi 0. Sedangkan jika semua input adalah bilangan biner 1, maka outputnya akan menjadi 1.</div></div><p style="text-align: center;"></p><p></p>Gerbang OR<div class="separator" style="clear: both;"><div class="separator" style="clear: both;"><div style="margin-left: 1em; margin-right: 1em; text-align: center;"></div></div>
<div style="text-align: justify;">
</div><div

style="text-align: justify;">Gerbang ini juga memerlukan dua input untuk menghasilkan satu output. Gerbang OR ini akan menghasilkan output 1 jika semua atau salah satu input merupakan bilangan biner 1. Sedangkan output akan menghasilkan 0 jika semua inputnya adalah bilangan biner 0.

<div style="text-align: justify;">

<u></u>

3. Dasar Teori [Kembali]</div><p class="MsoNormal" style="text-align: justify;">Carry lookahead adder adalah rangkaian adder yang lebih

cepat dalam melakukan penjumlahan biner dengan menggunakan konsep Carry

Generate dan Carry Propagate. CLA disebut sebagai penerus ripple carry adder.

Sirkuit CLA meminimalkan waktu tunda propagasi melalui implementasi sirkuit

kompleks yang terjadi selama penambahan ini dirancang dengan mengubah rangkaian

adder pembawa riak (ripple-carry adder) sedemikian rupa sehingga logika carry

penambah diubah menjadi logika dua tingkat.</p>

<p class="MsoNormal" style="text-align: justify;">Dalam kasus penjumlahan paralel, penambahan biner dari dua

angka dimulai ketika semua bit dari augend dan tambahan harus tersedia pada

saat yang sama untuk melakukan perhitungan. Dalam rangkaian penambah paralel,

keluaran pembawa dari setiap tahap penambah penuh dihubungkan ke masukan

pembawa dari tahap orde tinggi berikutnya, oleh karena itu disebut juga sebagai

penambah tipe pembawa riak.</p>

<p class="MsoNormal" style="text-align: justify;">Dalam rangkaian penambah seperti itu, tidak mungkin untuk

menghasilkan jumlah dan keluaran keluaran dari setiap tahap sampai pengangkutan

masukan terjadi. Sehingga akan terjadi tunda waktu yang cukup besar pada proses

penambahan , yang dikenal dengan , tunda propagasi carry. Dalam rangkaian

kombinasional apa pun, sinyal harus merambat melalui gerbang sebelum jumlah

keluaran yang benar tersedia di terminal keluaran.

[!\[\]\(b39c89771cd6fb2128a8c57aa7d97f9a_img.jpg\)](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEjCAUqMts-OWOyPmK5e7xxnwiypRzOrBeuK2oOSmZElwoUP96lemOuW3d9xNwKHZbPCwGZsrWPKV6B8A2cxFmHvSDRaQP3lovMS3cNkTxmu5JGrdeBujTVh9jD_5fpOeRHA0VKX00HueG4dgeW1cFRw1VwrKKFTbPQBw0x11Boq7AZP0EQAOzVWWsEsQ/s654/1.%204-bit-adder.jpg)

Pertimbangkan gambar di atas, di mana jumlah S4 dihasilkan

oleh penambah penuh yang sesuai segera setelah sinyal input diterapkan padanya.

Tetapi input carry C4 tidak tersedia pada nilai kondisi tunak akhirnya sampai

carry c3 tersedia pada nilai kondisi tunaknya. Demikian pula C3 tergantung pada

C2 dan C2 pada C1. Oleh karena itu, carry harus merambat ke semua tahapan agar

output S4 dan carry C5 menyelesaikan nilai kondisi tunak akhirnya.

Waktu propagasi sama dengan delay propagasi dari gerbang

tipikal dikalikan dengan jumlah level gerbang di sirkuit. Misalnya, jika setiap

tahap penambah penuh memiliki penundaan propagasi 20n detik, maka S4 akan

mencapai nilai akhir yang benar setelah 80n (20×4) detik. Jika kita

memperpanjang jumlah tahapan untuk menambahkan lebih banyak jumlah bit maka

situasi ini menjadi jauh lebih buruk.

Jadi kecepatan jumlah bit yang ditambahkan dalam penambah

paralel tergantung pada waktu propagasi carry. Namun, sinyal harus disebarkan

melalui gerbang pada waktu yang cukup untuk menghasilkan keluaran yang benar

atau diinginkan.

[!\[\]\(b538fe54c1f3a7343e37e85cc2d00497_img.jpg\)](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEiVB1LKd9P_viwuijXVnFEvOqIhzb2ChYhgKHqTMnHYm1HkcnPA9LeQIYtqzUZuAYZ2zvP9pGZuOSqi1ngmUOeHy_QIQZg9CQW0FKzQp_VE3oelPeaJsZlbiPa5Lexl0kyD8Y0bDsukb4NLrniqcFnY4P2_8g4u5zkR3gSEoTz4SR7MAXZF-nmTUNZOaw/s669/2.%20Full-adder1.jpg)

E3oeIPeaJsZlBlPa5Lexl0kyD8Y0bDsukb4NLrniqcFnY4P2_8g4u5zkR3gSEoTz4SR7MAXZF-nmTUNZOaw/w400-h176/2.%20Full-adder1.jpg" width="400" /></div>
<div class="separator" style="clear: both; text-align: center;"></div><div>
</div><p class="MsoNormal" style="background: white; line-height: normal; margin-bottom: 0cm; text-align: justify;">Pertimbangkan rangkaian adder penuh yang ditunjukkan

di atas dengan tabel kebenaran yang sesuai. Jika kita mendefinisikan dua variabel

sebagai carry generate Gi dan carry propagate Pi maka,<o:p></o:p></p><p class="MsoNormal" style="background: white; line-height: normal; margin-bottom: 0cm;"></p><div class="separator" style="clear: both; text-align: center;"></div><p></p><p class="MsoNormal" style="line-height: 106%; text-align: justify;">Jumlah keluaran dan keluaran carry

dapat dinyatakan dalam bentuk carry generate Gi dan carry propagat Pi sebagai<o:p></o:p></p><div class="separator" style="clear: both; text-align: center;"></div><p class="MsoNormal" style="line-height: 106%; text-align: justify;">di mana Gi menghasilkan carry

ketika kedua Ai dan Bi adalah 1 terlepas dari input carry. Pi dikaitkan dengan

propagasi carry dari Ci ke Ci + 1.<o:p></o:p></p>

<p class="MsoNormal" style="line-height: 106%; text-align: justify;">Fungsi carry output Boolean dari

setiap tahap dalam 4 tahap carry look-ahead adder dapat dinyatakan sebagai:<o:p></o:p></p><div

class="separator" style="clear: both; text-align: center;"></div><p class="MsoNormal" style="line-height: 106%; text-align:

justify;">Dari persamaan Boolean di atas kita

dapat mengamati bahwa C4 tidak harus menunggu C3 dan C2 untuk menyebar (propagate)

tetapi sebenarnya C4 disebarkan pada saat yang sama dengan C3 dan C2 . Karena

ekspresi Boolean untuk setiap keluaran carry adalah jumlah produk sehingga ini

dapat diimplementasikan dengan satu tingkat gerbang AND diikuti oleh gerbang

OR.<o:p></o:p></p>

<p class="MsoNormal" style="line-height: 106%; text-align: justify;">Implementasi tiga fungsi Boolean

untuk setiap output carry (C2 , C3 dan C4) untuk generator carry look-ahead

carry ditunjukkan pada gambar di bawah.<o:p></o:p></p><div class="separator" style="clear: both; text-align: center;"></div>
<p class="MsoNormal"

style="line-height: 106%; text-align: justify;">Analisis Kompleksitas Waktu :<o:p></o:p></p><p

class="MsoNormal" style="line-height: 106%; text-align: justify;">Kita bisa menganggap carry

look-ahead adder terdiri dari dua bagian.</p><p class="MsoNormal" style="line-height: 106%;

text-align: left;"></p><ol style="text-align: left;"><li style="text-align: justify;">Bagian yang

menghitung carry untuk

setiap bit.<li style="text-align: justify;">Bagian yang menambahkan bit input

dan carry untuk setiap posisi bit.

Keuntungan dan Kerugian dari Carry

Look-Ahead Adder :> Keuntungan > Penundaan propagasi berkurang.

Memberikan logika penambahan

tercepat.

Kekurangan Sirkuit penambah Carry Look-ahead

menjadi rumit karena jumlah variabel meningkat.

Sirkuit ini lebih mahal karena

melibatkan lebih banyak perangkat keras.

"georgia" , "times new roman" , serif;<u></u>

4. Percobaan [Kembali]</div>

Prosedur Percobaan<p style="text-align: center;">

</p><p></p><div style="text-align: justify;">1. Buka aplikasi proteus</div><div style="text-align: justify;">2. Pilih komponen yang dibutuhkan, pada rangkaian ini dibutuhkan gerbang XOR, Gerbang AND, dan gerbang OR.</div><div style="text-align: justify;">3. Rangkai setiap komponen menjadi rangkaian yang diinginkan</div><div style="text-align: justify;">4. Ubah spesifikasi komponen sesuai kebutuhan</div><div style="text-align: justify;">5. Jalankan simulasi rangkaian.</div><p></p>

<p></p><p style="text-align: center;">

</p><p></p><p></p><p></p>

Rangkaian Simulasi<div class="separator" style="clear: both;"><div class="separator" style="clear: both; text-align: center;">
</div>
</div>

<p></p><p style="text-align: center;">

Prinsip Kerja

Rangkaian carry look-ahead adder jika dilihat pada rangkaian tabel kebenaran menggunakan 2 gerbang XOR, 2 gerbang AND, dan 1 gerbang OR. Ada 3 kondisi yang dihasilkan ketika input diatur pada logicstate sedemikian rupa. Jika input A, B, C adalah 0, 0, 0 maka output di Ci+1 adalah 0 maka kondisi ini adalah no carry generate artinya tidak dihasilkan carry. Kemudian untuk kondisi selanjutnya jika input A, B, C adalah 0, 1, 1 maka output di Ci+1 adalah 1 maka kondisi ini adalah no carry propagate artinya tidak ada carry yang disebarkan. Kemudian untuk kondisi selanjutnya jika input A, B, C adalah 1, 1, 0 atau 1, 1, 1 maka output di Ci+1 adalah 1 maka kondisi ini adalah carry generate artinya carry yang dihasilkan.

5. Video [\[Kembali\]](#)

"georgia", "times new roman", serif;"georgia", "times new roman", serif;

6. Link Download [\[Kembali\]](#)

-

- Download rangkaian proteushttps://drive.google.com/file/d/1Xa7UNOMTCFFCUHkUujOEKcabZ_apBY26/view?usp=sharing>Klik
- Download video simulasi<https://drive.google.com/file/d/17wou-6YoALOGtfmuTuJH8XaCFy76oTx/view?usp=sharing>>Klik

[Download datasheet gerbang XOR](https://drive.google.com/file/d/1t--o1h3m5tcFdpzxd37OaftBKRHHmhle/view?usp=sharing)

[Download datasheet gerbang AND](https://drive.google.com/file/d/1f5yWHfz2Hc8ty1ebE1GliLLaMhtdXaUE/view?usp=sharing)

[Download datasheet gerbang OR](https://drive.google.com/file/d/1CmTCpZogcFXTBDMkL-4ohZiIJOODh06N/view?usp=sharing)

[Download HTML](#)

<div class="separator" style="clear: both; font-family: Georgia, Times New Roman, serif; text-align: center;">

</div>

<div class="separator" style="clear: both; font-family: Georgia, Times New Roman, serif; text-align: justify;">

</div>

<div class="separator" style="clear: both; font-family: Georgia, Times New Roman, serif; text-align: center;">

</div>

<div class="separator" style="clear: both; font-family: Georgia, Times New Roman, serif; text-align: justify;">

</div>

<div class="separator" style="clear: both; font-family: Georgia, Times New Roman, serif; text-align: center;">

</div>

<div class="separator" style="clear: both; font-family: Georgia, Times New Roman, serif; text-align: center;">

</div>

<div style="font-family: Georgia, "Times New Roman"; serif; text-align: center;">

[MENUJU AWAL]</div>

<center style="font-family: Georgia, "Times New Roman"; serif;">

</center>

<div class="separator" style="clear: both; text-align: center;">
</div>
<div class="separator" style="clear: both; text-align: center;">
</div>
</div>