

تمر التحميل من موقع الفريد في الفيزياء

الوحدة الخامسة: الدوائر المنطقية التتابعية Sequential Logic Circuits

محتويات الوحدة

تمهيد

أهداف الوحدة

1. الدوائر المنطقية التتابعية (Sequential Logic Circuits)

2. المراجيح (Flip Flops)

1-2 بناء المراجيح

2-2 المراجيح المتزامنة (Clocked Flip Flops)

3-2 مرجاح القائد-التابع (Master-Slave Flip Flop)

4-2 أطراف الدخل المباشر (Direct Inputs)

5-2 التزامن ثنائي الطور (Two-Phase Clocking)

3. المسجلات (Registers)

1-3 بناء المسجلات

2-3 الكتابة في المسجلات و القراءة منها (Write and Read Operations)

3-3 نقل البيانات ما بين المسجلات (Register-to-Register Transfer)

4-3 مسجلات الإزاحة (Shift Registers)

4. العدادات (Counters)

1-4 بناء العدادات

2-4 العد تصاعدياً (Up Counting)

3-4 العد تنازلياً (Down Counting)

4-4 العد في الاتجاهين (Up/Down Counting)

5-4 العد ضمن نطاق معين

6-4 العد بأي ترتيب

تمهيد

مرحباً بك عزيزي الدارس في الوحدة الخامسة من المقرر "أساسيات التصميم المنطقي". في ما سبق من المقرر تعاملنا مع دوائر منطقية ترابطية (Combinational Logic Circuits)، و فيها يعتمد خرج الدائرة فقط على القيم الحالية للدخل، أما في هذه الوحدة فسنتعرف على النوع الآخر من الدوائر المنطقية و هو الدوائر المنطقية التتابعية (Sequential Logic Circuits)، مثل المراجيح (Flip Flops) و المسجلات (Registers) و العدادات (Counters)، و فيها لا يعتمد الخرج فقط على القيم الحالية للدخل، و إنما يعتمد أيضاً على القيم السابقة للخرج.

أي أن هذا النوع من الدوائر له ذاكرة (Memory) تستطيع احتزان القيم السابقة لخرج الدائرة بحيث تستطيع التأثير على خرجها الحالي. و السبب في ظهور هذه القدرة التخزينية هو وجود تغذية مرتدة (Feedback) من خرج الدائرة إلى دخلها. حيث سنقوم في هذه الوحدة بعرض مبسط لأهم أنواع الدوائر المنطقية التتابعية و أكثرها شيوعاً في الاستخدام، و لن نتعرض لتصميم الدوائر المنطقية التتابعية بالتفصيل، كما فعلنا بالنسبة للدوائر المنطقية الترابطية، بل سنترك هذه الدراسة التفصيلية لمقرر آخر متقدم في التصميم المنطقي. نبدأ هنا بدراسة الوحدة الأساسية في بناء الدوائر المنطقية التتابعية و هي المراجيح (Flip Flops)، حيث نقوم بتوضيح بنائها و طريقة عملها و أنواعها المختلفة و استخدامات كل نوع. ثم ننتقل للمسجلات (Registers) حيث نقوم بتوضيح بنائها و كيفية الكتابة فيها و القراءة منها و كيفية نقل البيانات بينها، كما نتعرف على مسجلات الإزاحة (Shift Registers) بأنواعها المختلفة. و في نهاية الوحدة نتعرف على العدادات (Counters)، حيث نتعرف على بنائها و أنواعها المختلفة و استخداماتها.

أهداف الوحدة

- عزيزي الدارس، بعد دراسة هذه الوحدة ينبغي أن تكون قادراً على:
- التفريق ما بين الدوائر المنطقية الترابطية و الدوائر المنطقية التتابعية.
 - تصميم المراجيح بأنواعها المختلفة و توضيح طريقة عملها.
 - استخدام المراجيح في تصميم الأنظمة الرقمية.
 - استخدام مخططات التزامن في تحليل الدوائر المنطقية التتابعية.
 - تصميم المسجلات بأنواعها و استخدامها في الأنظمة الرقمية.
 - توضيح طريقة نقل البيانات بين المسجلات.
 - تصميم مسجلات الإزاحة.
 - تصميم العدادات بأنواعها المختلفة و شرح طريقة عملها و استخدامها في الأنظمة الرقمية.

1 - الدوائر المنطقية التتابعية (Sequential Logic Circuits)

تنقسم الدوائر المنطقية إلى نوعين؛ دوائر منطقية ترابطية (Combinational Logic Circuits) و دوائر منطقية تتابعية (Sequential Logic Circuits). سميت الدوائر المنطقية الترابطية بهذا الاسم نظراً إلى أن وظيفة الدائرة هي ربط متغيرات الدخل بعمليات منطقية لتوليد متغيرات الخرج، و بالتالي فإن خرج هذا النوع من الدوائر يعتمد فقط على القيم الحالية للدخل، فمتى ما تغير الدخل تبع ذلك تغير الخرج، و إذا لم يتغير الدخل يظل الخرج كما هو. و جميع الدوائر المنطقية التي تعاملنا معها في هذا المقرر حتى الآن، مثل الجوامع (Adders) و فاك الشفرة (Decoder) و المشفر (Encoder) و الدامج (Multiplexer) و المفرق (Demultiplexer)، هي دوائر منطقية ترابطية. أما الدوائر المنطقية التتابعية فلا يعتمد خرجها على القيم الحالية للدخل فقط و إنما يعتمد بالإضافة إلى ذلك على القيم السابقة للخرج، حيث أن هذا النوع من الدوائر له ذاكرة (Memory) تستطيع احتزان ماضي الدائرة بحيث يؤثر هذا الماضي

على الخرج الحالي. و السبب في ظهور القدرة التخزينية في الدوائر المنطقية التتابعية هو وجود تغذية مرتدة (Feedback)، حيث أن خرج الدائرة يتم أخذه عبر هذه التغذية المرتدة و إدخاله إلى الدائرة مرة أخرى مع متغيرات الدخل. و نظراً لوجود ماضي و حاضر في الدوائر المنطقية التتابعية نستطيع القول أن الزمن (Time) يدخل فيها كمتغير. و دخول الزمن كمتغير يتطلب وجود إشارة التزامن (Clock Signal) في الدوائر المنطقية التتابعية للقيام بدور تنسيقي و تنظيمي هام في النظام الرقمي. و الجدول التالي يلخص الفروقات ما بين الدوائر المنطقية الترابطية و الدوائر المنطقية التتابعية

| الدوائر المنطقية الترابطية | الدوائر المنطقية التتابعية | |
|---|---|-------------------------------|
| يعتمد على القيم الحالية للدخل فقط | يعتمد على القيم الحالية للدخل و على القيم السابقة للخرج | 1. الخرج |
| ليس لها ذاكرة | لها ذاكرة | 2. الذاكرة (Memory) |
| لا توجد بها تغذية مرتدة | توجد بها تغذية مرتدة | 3. التغذية المرتدة (Feedback) |
| لا يدخل الزمن فيها كمتغير | يدخل الزمن فيها كمتغير | 4. الزمن |
| لا توجد بها إشارة تزامن | توجد بها إشارة تزامن | 5. إشارة التزامن (Clock) |
| الجوامع، فاك الشفرة، المشفر، الدامج، المفرق | المراجع، المسجلات، العدادات | 6. أمثلة |

2- المراجع (Flip Flops)

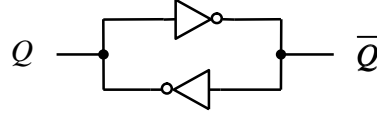
المراجع (Flip Flop) عبارة عن دائرة منطقية تتابعية لها القدرة على تخزين خانة ثنائية واحدة (1-bit) فقط من البيانات. و يطلق عليه باللغة العربية أيضاً تسمية القلاب أو النطاظ، و لكن سنستخدم هنا تسمية المراجع نظراً لفصاحتها و لأدائها للمعنى المطلوب بدقة أكبر. حيث أن للمراجع حالتين (two states) يتأرجح بينهما، أي ينتقل من إحداهما إلى الأخرى تحت تأثير متغيرات الدخل. تسمى الحالة الأولى للمراجع و التي يكون محتفظاً فيها بالقيمة المنطقية 1 بحالة SET، في حين تسمى الحالة الأخرى و التي يكون محتفظاً فيها بالقيمة المنطقية 0 بحالة RESET أو CLEAR. هذا و يعتبر المراجع وحدة البناء الأساسية لجميع الدوائر المنطقية التتابعية.

1-2 بناء المراجع

من الممكن أن يتم بناء المراجع باستخدام العواكس المنطقية أو باستخدام بوابات NOR أو باستخدام بوابات NAND.

مرجاح من العواكس المنطقية:

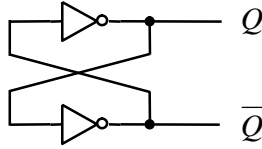
يتكون أبسط أنواع المراجيح من عاكسين منطقيين يقوم خرج كل منهما بتغذية دخل الآخر، كما هو موضح بالشكل التالي



يسمى الطرف Q بالخرج غير المعكوس للمرجاح، في حين يسمى الطرف \bar{Q} بالخرج المعكوس.

لتخزين قيمة معينة في المرجاح نقوم بتسليط الجهد الكهربائي الممثل لتلك القيمة من مصدر خارجي على الطرف Q لفترة زمنية قصيرة جداً (الفترة الزمنية اللازمة لظهور خرج العاكس المنطقي الثاني)، ثم نقوم بإزالة مصدر الدخل الخارجي، فيظل المرجاح محتفظاً بتلك القيمة المخزنة به ما دامت تغذية بواباته المنطقية بالقدر الكهربي مستمرة، و يفقد القيمة المخزنة به عند إنقطاع تلك التغذية.

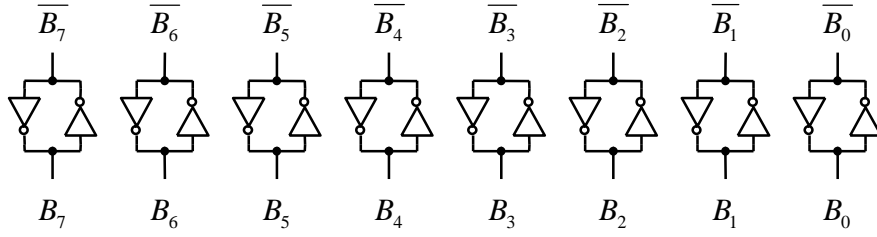
يمكن رسم دائرة المرجاح البسيط المكون من عاكسين منطقيين بالصورة التالية



نلاحظ هنا وجود التغذية المرتدة (Feedback) من طرفي الخرج للعاكسين المنطقيين إلى طرفي الدخل لهما.

يطلق على هذا المرجاح تسمية Static Latch. و مصطلح Static في الدوائر المنطقية يشير إلى غياب إشارة التزامن (Clock)، و المصطلح العكسي Dynamic يشير إلى وجود تلك الإشارة. و غياب إشارة التزامن هنا يعني عدم إمكانية تغير حالة الدائرة بمرور الزمن فقط، أي أن القيمة المخزنة في المرجاح ستظل كما هي حتى يتم استبدالها بقيمة أخرى. يستخدم هذا المرجاح كوحدة بناء أساسية في نوع من أنواع الذاكرة (Memory) يسمى Static RAM أو SRAM، كما سيتم توضيحه بالتفصيل في الوحدة التالية من المقرر.

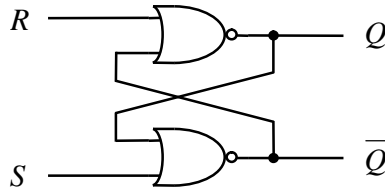
ذكرنا أن المرجاح له القدرة على تخزين خانة ثنائية واحدة (1-bit) فقط من البيانات، فلتخزين معلومة مكونة من مجموعة من الخانات الثنائية نحتاج لعدد من المراجيح بعدد الخانات الثنائية (bits) المطلوب تخزينها، كما هو موضح بالشكل التالي



و تسمى مجموعة المراجيح المستخدمة في تخزين معلومة مكونة من عدد من الخانات الثنائية بالمسجل (Register).

مرجاح من بوابات NOR:

بما أن بوابة NOR يمكن أن تعمل عمل العاكس المنطقي، لذلك يمكن استخدامها في بناء المراجيح كما هو موضح أدناه

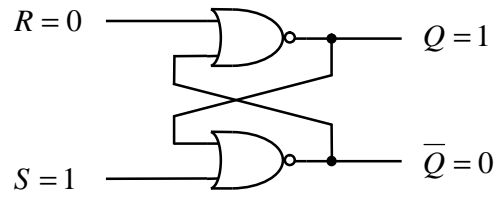


لاحظ أن وجود أكثر من طرف دخل لبوابة NOR سمح لنا بإضافة أطراف أخرى للمرجاح هي أطراف التحكم S و R ، والتي يمكن عن طريقها التحكم في حالة المرجاح. ف S هو إختصار لكلمة SET و هي حالة المرجاح التي تكون فيها القيمة المنطقية 1 مخزنة فيه، و R هو إختصار لكلمة RESET و هي حالة المرجاح التي تكون فيها القيمة المنطقية 0 مخزنة فيه. أي أن المرجاح يكون في حالة SET إذا كانت القيمة المخزنة فيه هي 1، و يكون في حالة RESET إذا كانت القيمة المخزنة فيه هي 0، علماً بأن القيمة المخزنة في المرجاح هي القيمة التي تظهر في طرف الخرج غير المعكوس Q .

يطلق على هذا المرجاح تسمية مرجاح SET/RESET أو مرجاح SR (SR Flip Flop) إختصاراً.

• إجراء عملية SET للمرجاح:

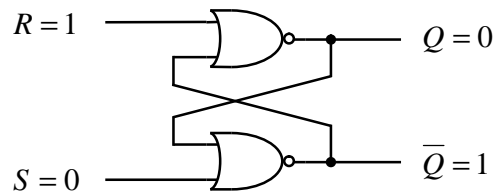
لإجراء عملية SET للمرجاح نضع القيمة المنطقية 1 في الطرف المقابل للعملية المطلوب إجراؤها، أي الطرف S ، و نضع القيمة المنطقية 0 في الطرف الآخر، أي الطرف R ، كما هو موضح أدناه



لاحظ أن دخول الـ 1 إلى بوابة NOR الموجودة بالأسفل يحدد خرجها بـ $\bar{Q} = 0$. و بمعلومية قيمة \bar{Q} و قيمة R يمكن تحديد خرج بوابة NOR الموجودة بالأعلى بـ $Q = 1$.

- إجراء عملية RESET للمرجاح:

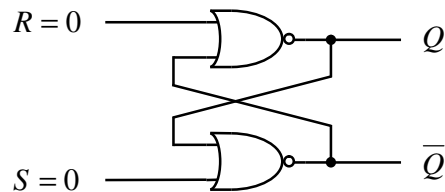
لإجراء عملية RESET للمرجاح نضع القيمة المنطقية 1 في الطرف المقابل للعملية المطلوب إجراؤها، أي الطرف R ، و نضع القيمة المنطقية 0 في الطرف الآخر، أي الطرف S ، كما هو موضح أدناه



لاحظ أن دخول الـ 1 إلى بوابة NOR الموجودة بالأعلى يحدد خرجها بـ $Q = 0$. و بمعلومية قيمة Q و قيمة S يمكن تحديد خرج بوابة NOR الموجودة بالأسفل بـ $\bar{Q} = 1$.

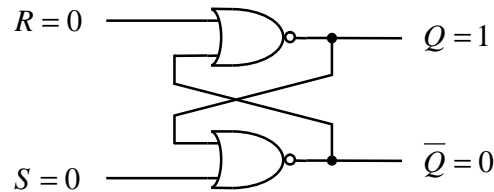
يوجد احتمالي دخل آخرين للطرفين S و R ، الاحتمال الأول هو $S = 0$ و $R = 0$ ، و الاحتمال الثاني هو $S = 1$ و $R = 1$. المطلوب الآن إيجاد حالة المرجاح لكل احتمال دخل منهما

- احتمال الدخل $S = 0$ و $R = 0$:



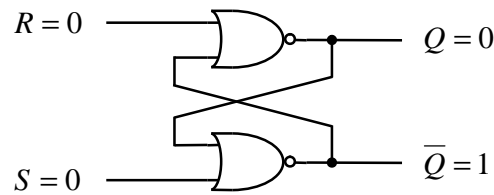
لا نستطيع هنا تحديد حالة المرحاح الجديدة دون معرفة حالته السابقة، لأنه إذا كانت القيمة الموجودة على أحد طرفي الدخول لبوابة NOR هي 0 فلا يمكن تحديد خرجها دون معرفة القيمة الموجودة على طرف الدخول الآخر.

أولاً: إذا كان المرحاح في حالة SET، أي أن $Q = 1$ و $\bar{Q} = 0$



نجد أن الحالة الجديدة للمرحاح هي أيضاً حالة SET، أي أن المرحاح احتفظ بحالته السابقة.

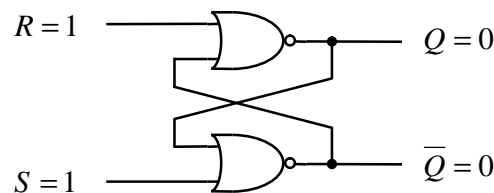
ثانياً: إذا كان المرحاح في حالة RESET، أي أن $Q = 0$ و $\bar{Q} = 1$



نجد أن الحالة الجديدة للمرحاح هي أيضاً حالة RESET، أي أن المرحاح احتفظ بحالته السابقة.

و عليه نستنتج أنه في حالة الدخول $S = 0$ و $R = 0$ يحتفظ المرحاح بحالته السابقة.

• احتمال الدخول $S = 1$ و $R = 1$:



يؤدي احتمال الدخول هذا إلى جعل كلا طرفي الخرج Q و \bar{Q} مساويين 0، وهو أمر غير مسموح به. كما أنه عند عودة القيمة الموضوعة على طرفي الدخول S و R من 1 إلى 0 في وقت واحد فإن حالة المرحاح تكون غير محددة، أي لا يمكن التكهّن بها، لأنها تعتمد على أي الطرفين S و R تغير قبل الآخر. لذلك فإن احتمال الدخول $S = 1$ و $R = 1$ غير مستخدم أو غير مسموح به (Invalid).

هذا و يمكن تلخيص النتائج السابقة في جدول الصواب (Truth Table) التالي

| S | R | Q_{n+1} | |
|-----|-----|-----------|-------|
| 0 | 0 | Q_n | Keep |
| 0 | 1 | 0 | RESET |
| 1 | 0 | 1 | SET |
| 1 | 1 | Invalid | |

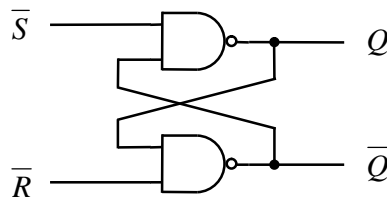
حيث Q_{n+1} هي الحالة الجديدة للمرحاح، و Q_n هي الحالة السابقة للمرحاح.

كما يمكن أن نقوم في جدول الصواب بإدراج الخرج المعكوس \bar{Q} إضافة إلى الخرج غير المعكوس Q ، و ذلك لتوضيح ما يحدث في حالة الدخول $S = 1$ و $R = 1$ ، كما هو موضح أدناه

| S | R | Q_{n+1} | \bar{Q}_{n+1} | |
|-----|-----|-----------|-----------------|---------|
| 0 | 0 | Q_n | \bar{Q}_n | Keep |
| 0 | 1 | 0 | 1 | RESET |
| 1 | 0 | 1 | 0 | SET |
| 1 | 1 | 0 | 0 | Invalid |

مرحاح من بوابات NAND:

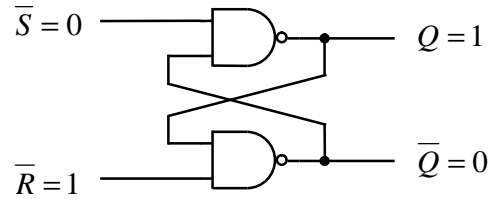
بما أن بوابة NAND، مثلها في ذلك مثل بوابة NOR، يمكن أن تعمل عمل العاكس المنطقي، لذلك يمكن استخدامها في بناء المراجيح كما هو موضح أدناه



المرجاح هنا أيضاً عبارة عن مرجاح SET/RESET أو مرجاح SR (SR Flip Flop)، إلا أن دخله نشط منخفض (Active Low)، أي أن العملية المطلوبة يتم إجراؤها بوضع 0 في الطرف المقابل لها.

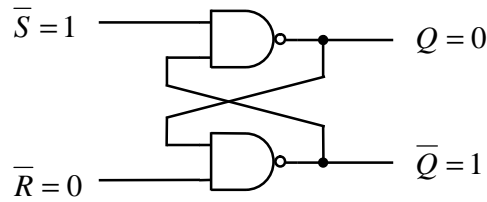
- إجراء عملية SET للمرجاح:

يتم ذلك بجعل $\bar{S} = 0$ و $\bar{R} = 1$ ، كما هو موضح أدناه



- إجراء عملية RESET للمرجاح:

يتم ذلك بجعل $\bar{S} = 1$ و $\bar{R} = 0$ ، كما هو موضح أدناه



هذا ويمكن بسهولة إثبات أن الدخل $\bar{S} = 1$ و $\bar{R} = 1$ يؤدي لاحتفاظ المرجاح بحالته السابقة، و الدخل $\bar{S} = 0$ و $\bar{R} = 0$ يؤدي إلى جعل كلا الخرجين Q و \bar{Q} مساويين للقيمة 1.

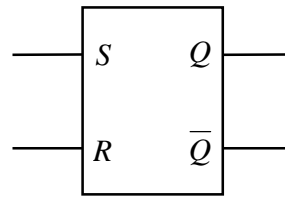
و يمكن تلخيص هذه النتائج في جدول الصواب التالي

| \bar{S} | \bar{R} | Q_{n+1} | |
|-----------|-----------|-----------|-------|
| 0 | 0 | Invalid | |
| 0 | 1 | 1 | SET |
| 1 | 0 | 0 | RESET |
| 1 | 1 | Q_n | Keep |

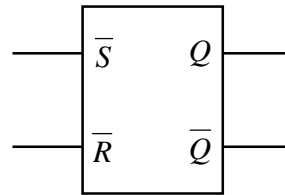
أو

| \bar{S} | \bar{R} | Q_{n+1} | \bar{Q}_{n+1} | |
|-----------|-----------|-----------|-----------------|---------|
| 0 | 0 | 1 | 1 | Invalid |
| 0 | 1 | 1 | 0 | SET |
| 1 | 0 | 0 | 1 | RESET |
| 1 | 1 | Q_n | \bar{Q}_n | Keep |

و الشكل التالي يوضح المخطط المنطقي (Logic Diagram) لمراجح SR



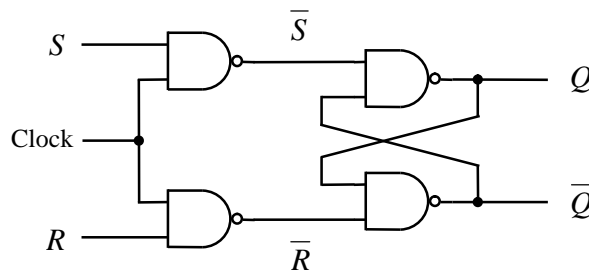
دخـل نشـط مـرتـفـع
(Active High Inputs)



دخـل نشـط مـنخـفـض
(Active Low Inputs)

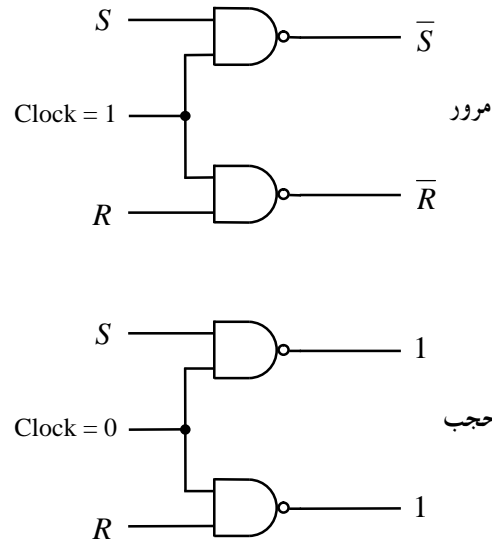
2-2 المراجيح المتزامنة (Clocked or Gated Flip Flops)

المراجيح المتزامنة (Clocked Flip Flop) تدخـل عـلـيـه إـشـارة تـسـمـى إـشـارة التـزامـن (Clock Signal) أو Clock إختصاراً. و تدخـل إـشـارة التـزامـن عـلـى مـراجـح SR بالطريقة الموضحة بالشكل التالي



و يطلق على المرحاح هنا تسمية مرجاح SR المتزامن (Clocked SR Flip Flop).

و إشارة التزامن (Clock) تشبه في عملها إلى حد كبير إشارة السماح (Enable)، فإذا كانت إشارة التزامن مرتفعة (High)، أي مساوية 1، تمر الإشارتان S و R إلى المرحاح و يستجيب لهما بالصورة المعتادة، أما إذا كانت إشارة التزامن منخفضة (Low)، أي مساوية 0، فيتم حجب الإشارتين S و R عن المرحاح و يظل المرحاح محتفظاً بحالته السابقة. كما هو موضح أدناه



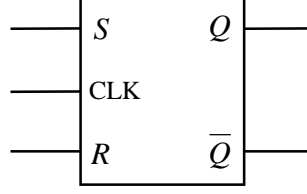
و في ما يلي جدول الصواب لمرجاح SR المتزامن

| C | S | R | Q_{n+1} | \bar{Q}_{n+1} | |
|-----|-----|-----|-----------|-----------------|---------|
| 0 | × | × | Q_n | \bar{Q}_n | Keep |
| 1 | 0 | 0 | Q_n | \bar{Q}_n | Keep |
| 1 | 0 | 1 | 0 | 1 | RESET |
| 1 | 1 | 0 | 1 | 0 | SET |
| 1 | 1 | 1 | 1 | 1 | Invalid |

حيث المتغير C يمثل قيمة إشارة التزامن (Clock).

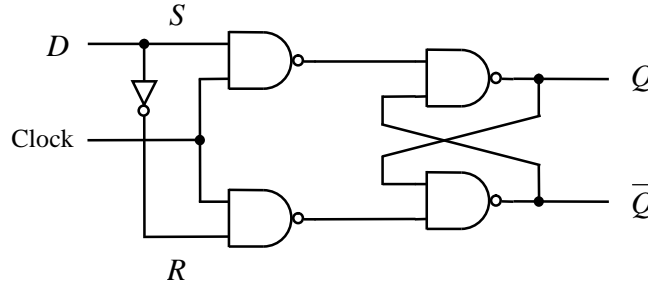
لاحظ أن دخل مرجاح SR المتزامن نشط مرتفع (Active High).

الشكل التالي يمثل المخطط المنطقي لمرجاح SR المتزامن



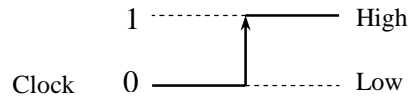
مرجاح D (D Flip Flop)

D هنا إختصار لكلمة Data، أي أن الأسم الكامل للمرجاح هو Data Flip Flop. و مرجاح D عبارة عن مرجاح SR متزامن تم ربط طرفي الدخول S و R له في طرف دخل واحد هو D باستخدام عاكس منطقي، كما هو موضح بالشكل التالي

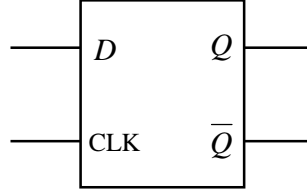


فإذا وضعنا القيمة المنطقية 0 في الطرف D يكون $S = 0$ و $R = 1$ فتحدث عملية RESET للمرجاح، أي يتم إختزان القيمة 0 فيه. وإذا وضعنا القيمة المنطقية 1 في الطرف D يكون $S = 1$ و $R = 0$ فتحدث عملية SET للمرجاح، أي يتم إختزان القيمة 1 فيه. أي أن القيمة التي يتم وضعها على الطرف D يتم إختزانها داخل المرجاح.

لاحظ ارتباط إنتقال القيمة الموضوعه على الطرف D و إختزانها داخل المرجاح بإشارة التزامن (Clock)، حيث تنتقل القيمة إلى داخل المرجاح و تحتزن في اللحظة التي تتغير فيها إشارة التزامن من Low إلى High، كما هو موضح أدناه



و في ما يلي المخطط المنطقي و جدول الصواب لمرجاح D



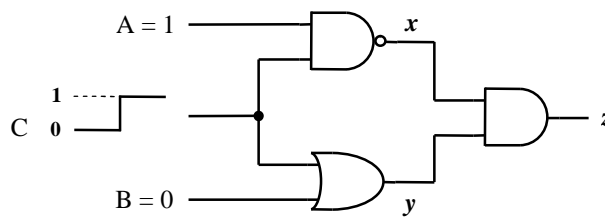
| C | D | Q_{n+1} |
|---|----------|-----------|
| 0 | \times | Q_n |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

هذا و يطلق على مرجاح D أيضاً تسمية Dynamic Latch، و يستخدم أساساً في بناء المسجلات (Registers).

أهمية التزامن (Timing)

لاحظنا ظهور إشارة التزامن (Clock) في الدوائر المنطقية المتتابعة (Sequential Logic Circuits)، و لم نلاحظها من قبل في الدوائر المنطقية الترابطية (Combinational Logic Circuits)، فما أهمية التزامن بالنسبة لدوائر المنطقية المتتابعة؟

سنقوم بتوضيح أهمية التزامن باستخدام الدائرة البسيطة التالية



للدائرة ثلاثة متغيرات دخل هي A و B و C. المتغير A ثابت دوماً في القيمة 1، و المتغير B ثابت دوماً في القيمة 0، أما المتغير C فتتغير قيمته في لحظة معينة من 0 إلى 1. و المطلوب إيجاد خرج الدائرة z.

- كدائرة منطقية ترابطية:

إذا تعاملنا مع الدائرة كدائرة منطقية ترابطية و لم نأخذ عامل الزمن في الاعتبار نجد أن

$$x = \overline{AC}$$

$$y = B + C$$

$$z = xy = \overline{AC} (B + C)$$

و بوضع $B=0$ و $A=1$ يكون

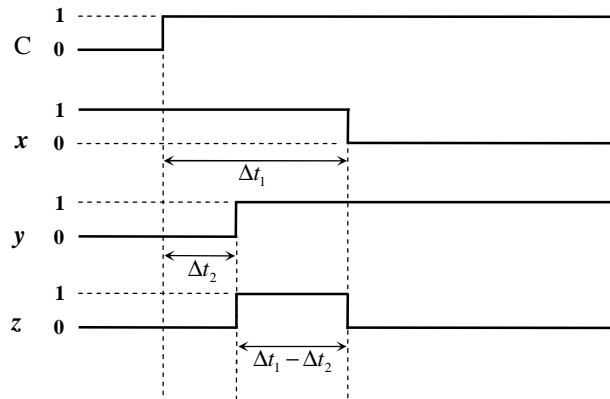
$$z = \overline{1 \cdot C} (0 + C) = \overline{C} C = 0$$

أي أن الخرج z يساوي 0 على الدوام بغض النظر عن التغير الحادث في قيمة المتغير C .

- كدائرة منطقية تنابعية:

إذا تعاملنا مع الدائرة كدائرة منطقية تنابعية فلا بد من أخذ عامل الزمن في الاعتبار، فلكل بوابة منطقية زمن تأخر إنتقال (Propagation Delay) هو عبارة عن الفترة الزمنية التي تمضي ما بين تسليط الدخل على البوابة و ظهور الإستجابة في خرجها. و زمن تأخر الإنتقال للبوابات المنطقية صغير جداً و يقاس بالنانوثانية (ns). و يختلف زمن تأخر الإنتقال من بوابة إلى أخرى، كما يختلف لنفس البوابة باختلاف الظروف المحيطة بها مثل درجة الحرارة. أي أنه من الصعب تحديد زمن تأخر الإنتقال لبوابة معينة بدقة.

في الدائرة أعلاه نفترض أن زمن تأخر الإنتقال لبوابة NAND هو Δt_1 ، و زمن تأخر الإنتقال لبوابة OR هو Δt_2 ، كما نفترض أن $\Delta t_1 > \Delta t_2$. و عليه فإن التغير في خرج بوابة NAND، أي التغير في قيمة المتغير x ، يحدث بعد التغير في دخلها، أي التغير في قيمة المتغير C ، بزمن مقداره Δt_1 . و التغير في خرج بوابة OR، أي التغير في قيمة المتغير y ، يحدث بعد التغير في دخلها، أي التغير في قيمة المتغير C ، بزمن مقداره Δt_2 . كما هو موضح بالشكل التالي



نلاحظ هنا أن الخرج z قد أصبح مساوياً 1 لفترة زمنية قصيرة جداً تساوي $\Delta t_1 - \Delta t_2$ ، وهو أمر غير متوقع. و يسمى مثل هذا الخرج غير المتوقع، و الناتج عن اختلاف زمن تأخر الانتقال للبوابات المنطقية، بالـ Hazard. و من الواضح أنه من الصعب جداً التنبؤ بمكان أو زمان ظهور هذه الـ Hazards في الدوائر المنطقية، حيث أن ذلك يتطلب تحليلاً غاية في الدقة للدائرة المنطقية، يؤخذ فيه في الاعتبار زمن تأخر الانتقال لكل بوابة منطقية. و إن كان هذا ممكناً للدائرة البسيطة أعلاه فإنه يكاد يكون مستحيلاً بالنسبة للدوائر المعقدة.

السؤال الآن هو ما تأثير هذه الـ Hazards على الدائرة المنطقية؟

- بالنسبة للدوائر المنطقية الترابطية:
نظراً لظهور الـ Hazards لفترة زمنية غاية في القصر و تلاشيها بعد ذلك فإنها تكاد أن تمر دون أن تلاحظ، و لا يكون لها بالتالي أي تأثير على الدائرة المنطقية.
- النسبة للدوائر المنطقية التتابعية:
يوجد هنا احتمال أن يقوم أحد المراجيح بالتقاط الـ Hazard أثناء فترة ظهورها القصيرة و تخزينها. عند ذلك لا يعود تأثير الـ Hazards على الدائرة المنطقية تأثيراً وقتياً و إنما يصبح تأثيراً دائماً.

إذن كيف نتلافى تأثير الـ Hazards على الدوائر المنطقية التتابعية؟

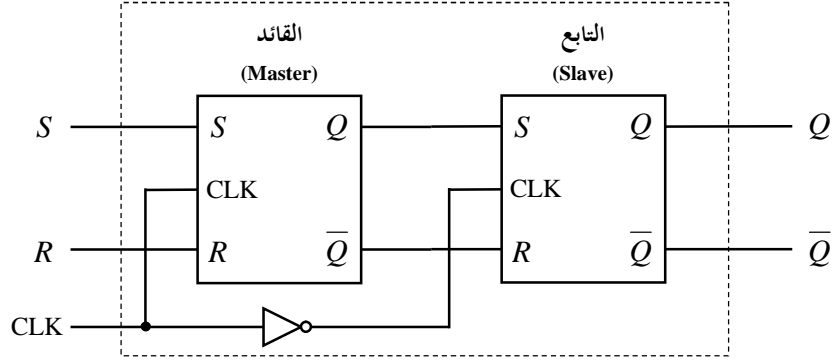
لتلافي تأثير الـ Hazards على الدائرة المنطقية التتابعية يكفي الإنتظار لفترة زمنية كافية لتلاشي الـ Hazards قبل قراءة خرج الدائرة، بحيث نضمن أن ذلك الخرج خالي من الـ Hazards. و هنا يأتي دور إشارة التزامن (Clock) التي تقوم بتنظيم فترات الإنتظار هذه. فإشارة التزامن عبارة عن إشارة تتغير قيمتها بانتظام ما بين 0 و 1، كما هو موضح بالشكل التالي



فالفترات التي تكون فيها إشارة التزامن منخفضة (Low) هي عبارة عن فترات إنتظار لضمان تلاشي الـ Hazards.

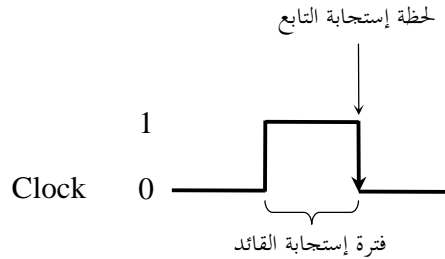
3-2 مرجاح القائد-التابع (Master-Slave Flip Flop)

يتكون مرجاح القائد-التابع من مرجاحي SR متزامنين متصلين ببعضهما البعض بحيث يغذي خرج أولهما دخل الثاني، كما هو موضح بالشكل التالي



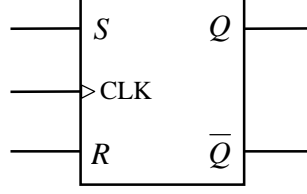
يسمى المرجاح الأول بالقائد (Master) و يسمى المرجاح الثاني بالتابع (Slave).

لاحظ أن إشارة التزامن (Clock) تدخل مباشرة إلى مرحلة القائد في حين تدخل معكوسة إلى مرحلة التابع، و معنى هذا أن إستجابة المرجاحين لا تتم في وقت واحد. فعندما تكون إشارة التزامن مرتفعة (High) يستجيب مرجاح القائد للدخل S و R ، في حين يكون مرجاح التابع في ذلك الوقت مغلقاً و محتفظاً بحالته السابقة، و في اللحظة التي تهبط فيها إشارة التزامن من High إلى Low ينغلق مرجاح القائد و تنتقل حالته إلى مرجاح التابع و تظهر في الخرج. كما هو موضح بالشكل التالي



أي أن مرجاح القائد-التابع يستجيب لأي تغير يحدث في طرفي الدخل S و R طالما كانت إشارة التزامن (Clock) مرتفعة (High)، و تظهر الإستجابة في خروجه لحظة هبوط إشارة التزامن من High إلى Low. و الإستجابة التي تظهر في الخرج هنا هي آخر حالة للمرجاح مباشرة قبل هبوط إشارة التزامن. و يظل خرج المرجاح ثابتاً بعد هبوط إشارة التزامن و ذلك حتى الهبوط الذي يليه.

و في ما يلي المخطط المنطقي و جدول الصواب لمرجاح القائد-التابع

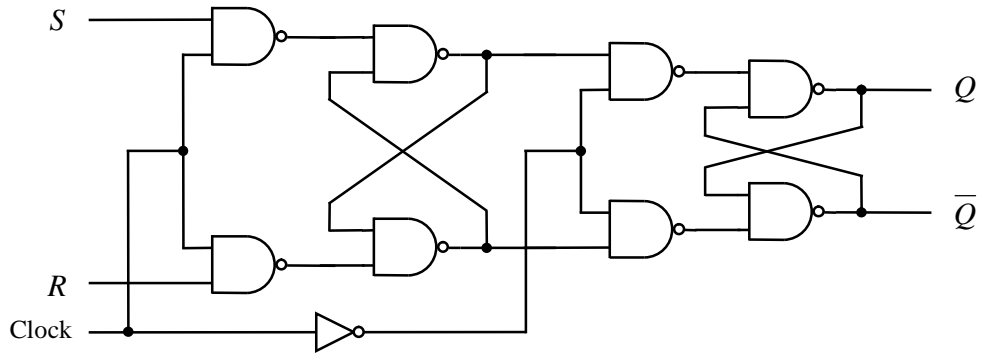


| C | S | R | Q_{n+1} | \bar{Q}_{n+1} | |
|-----|----------|----------|-----------|-----------------|---------|
| 0 | \times | \times | Q_n | \bar{Q}_n | Keep |
| 1 | 0 | 0 | Q_n | \bar{Q}_n | Keep |
| 1 | 0 | 1 | 0 | 1 | RESET |
| 1 | 1 | 0 | 1 | 0 | SET |
| 1 | 1 | 1 | 1 | 1 | Invalid |

لاحظ في المخطط المنطقي المثلث الصغير الموضوع عند طرف الدخل لإشارة التزامن (Clock) و الذي يدل على أن دخل المرجاح ينشط مع الحافة الصاعدة لنبضة التزامن، أي لحظة إنتقال إشارة التزامن من Low إلى High.

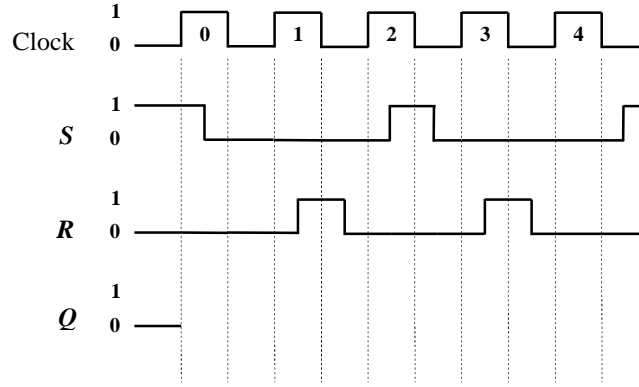
لاحظ أيضاً أن جدول الصواب لمرجاح القائد-التابع هو نفسه جدول الصواب لمرجاح SR المتزامن، أي أن كلا المرجاحين يستجيبان للدخل S و R بنفس الطريقة، و لكن الفرق بينهما يكون في لحظة ظهور الإستجابة في الخرج. ففي مرجاح SR المتزامن تظهر الإستجابة في الخرج فور حدوث التغير في الدخل، ما دامت إشارة التزامن (Clock) مرتفعة، أما في مرجاح القائد-التابع فلا تظهر الإستجابة في الخرج إلا لحظة هبوط نبضة التزامن من High إلى Low.

الشكل التالي يوضح الدائرة المنطقية لمرجاح القائد-التابع



مخططات التزامن (Timing Diagrams)

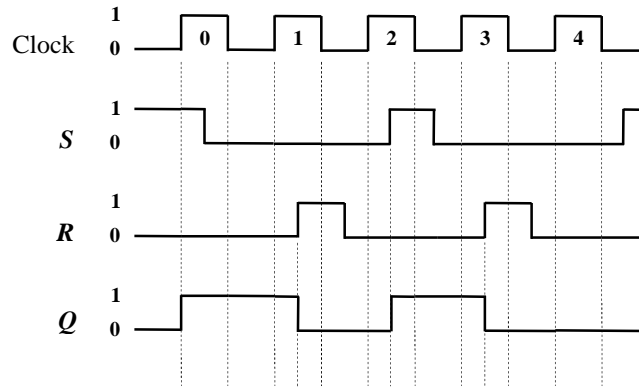
نظراً إلى أن الزمن (Time) يدخل كمتغير في الدوائر المنطقية التتابعية فلا بد من وسيلة لمتابعة التغير الذي يحدث في حالة الدائرة مع الزمن. هذه الوسيلة هي مخطط التزامن (Timing Diagram). فمخطط التزامن يوضح التغير الذي يحدث في متغيرات الدخل و الخرج للدائرة المنطقية مع الزمن. على سبيل المثال يوضح الشكل التالي مخطط تزامن معطى فيه الإشارات الداخلة إلى مرجاح SR متزامن، و هي إشارة التزامن (Clock) و متغيري الدخل S و R ، و معطى فيه أيضاً الحالة الابتدائية للمرجاح و هي حالة RESET، و مطلوب إيجاد خرج المرجاح Q



(ملاحظة: لتسهيل متابعة الشرح قمنا بترقيم نبضات التزامن)

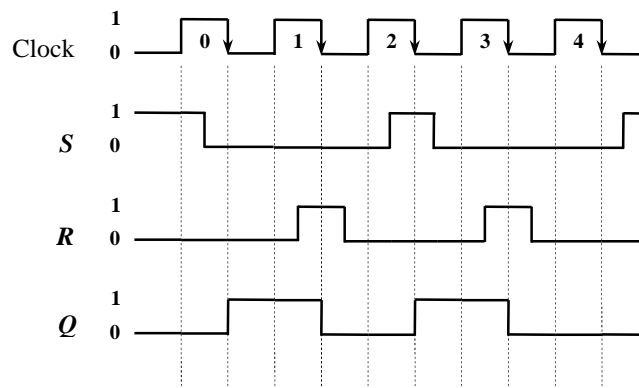
نعلم أن مرجاح SR المتزامن يستجيب للدخل S و R فقط عندما تكون إشارة التزامن مرتفعة (High)، و يظل محتفظاً بآخر حالة وصل إليها عندما تكون إشارة التزامن منخفضة (Low). لذلك ننظر إلى قيم الدخل S و R و التغير الذي يحدث فيها في كل نبضة من نبضات التزامن، منذ بداية النبضة و حتى نهايتها، و نوضح الاستجابة لهذه التغيرات على الخرج Q . فمع بداية النبضة رقم 0 نجد أن $S = 1$ و $R = 0$ مما يؤدي إلى حدوث عملية SET للمرجاح، و عند منتصف النبضة رقم 0 تقريباً تتحول قيمة S من 1 إلى 0، أي يصبح دخل المرجاح هو $S = 0$ و $R = 0$ و يؤدي هذا لاحتفاظ المرجاح بحالته أي حالة SET، و يستمر ذلك حتى نهاية النبضة رقم 0. و ما بين النبضة رقم 0 و النبضة رقم 1 لا يحدث أي تغير في حالة المرجاح نظراً إلى أن إشارة التزامن منخفضة (Low). مع بداية النبضة رقم 1 نجد أن $S = 0$ و $R = 0$ مما يؤدي إلى احتفاظ المرجاح بحالته، أي حالة SET، و ذلك حتى منتصف النبضة رقم 1 تقريباً، حيث تتحول قيمة R من 0 إلى 1، أي يصبح دخل المرجاح هو $S = 0$ و $R = 1$ و يؤدي هذا لحدوث عملية RESET للمرجاح، و يستمر ذلك حتى نهاية النبضة رقم 1. ما بين النبضة رقم 1 و النبضة رقم 2 لا يحدث أي تغير في حالة المرجاح. مع بداية النبضة رقم 2 نجد أن $S = 0$ و $R = 0$ مما يؤدي إلى احتفاظ المرجاح بحالته، أي حالة RESET، و ذلك حتى منتصف النبضة رقم 2 تقريباً، حيث تتحول قيمة S من 0 إلى 1، أي يصبح دخل المرجاح هو $S = 1$ و $R = 0$ و يؤدي هذا لحدوث عملية SET للمرجاح، و يستمر ذلك حتى نهاية النبضة رقم 2. ما بين النبضة رقم 2 و النبضة رقم 3 لا يحدث أي تغير في حالة المرجاح. مع بداية النبضة رقم 3 نجد

أن $S = 0$ و $R = 0$ مما يؤدي إلى احتفاظ المرحاح بحالته، أي حالة SET، و ذلك حتى منتصف النبضة رقم 3 تقريباً، حيث تتحول قيمة R من 0 إلى 1، أي يصبح دخل المرحاح هو $S = 0$ و $R = 1$ و يؤدي هذا لحدوث عملية RESET للمرحاح، و يستمر ذلك حتى نهاية النبضة رقم 3. ما بين النبضة رقم 3 و النبضة رقم 4 لا يحدث أي تغير في حالة المرحاح. في بداية النبضة رقم 4 نجد أن $S = 0$ و $R = 0$ مما يؤدي لاحتفاظ المرحاح بحالته، أي حالة RESET، و يستمر ذلك حتى نهاية النبضة. و بعد نهاية النبضة رقم 4 لا يحدث أي تغير في حالة المرحاح. كما هو موضح بالشكل التالي



ماذا لو كان مطلوباً إكمال نفس مخطط التزامن و لكن لمرحاح من نوع القائد-التابع (Master-Slave Flip Flop)؟

نعلم أن مرحاح القائد-التابع يستجيب للدخل S و R بنفس الطريقة التي يستجيب بها مرحاح SR المتزامن، إلا أن استجابة مرحاح القائد التابع لا تظهر في خرجه إلا لحظة هبوط نبضة التزامن من High إلى Low. و عليه يكون شكل مخطط التزامن لمرحاح القائد-التابع هو



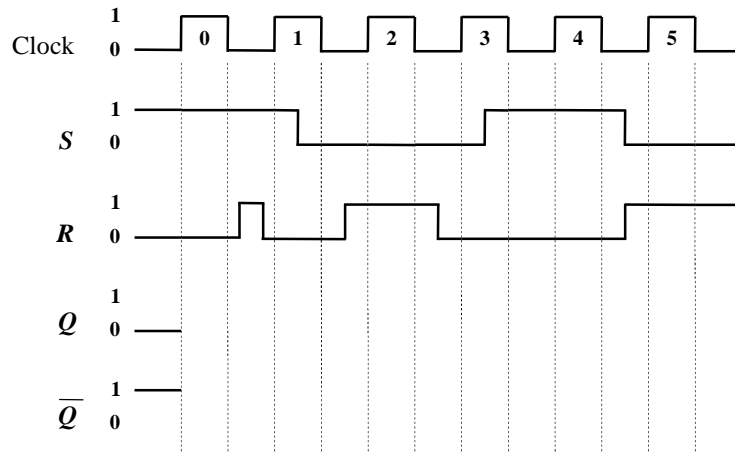
لاحظ أن الإستجابة لعملية SET التي حدثت في بداية النبضة رقم 0 لم تظهر في خرج المرحاح إلا في نهاية النبضة، لحظة الهبوط من High إلى Low. و لم يحدث أي تغيير في خرج المرحاح بين ذلك الهبوط و الهبوط الذي يليه في نهاية النبضة رقم 1. ففي منتصف النبضة رقم 1 حدثت عملية RESET و لكن لم تظهر الإستجابة لها في الخرج إلا

لحظة الهبوط في نهاية النبضة رقم 1. و لم يحدث أي تغيير في خرج المرجاح بين ذلك الهبوط و الهبوط التالي في نهاية النبضة رقم 2. ففي منتصف النبضة رقم 2 حدثت عملية SET و لكن لم تظهر في الخرج إلا لحظة الهبوط في نهاية النبضة رقم 2. و لم يحدث أي تغيير في خرج المرجاح بين ذلك الهبوط و الهبوط التالي في نهاية النبضة رقم 3. ففي منتصف النبضة رقم 3 حدثت عملية RESET و لكن لم تظهر في الخرج إلا لحظة الهبوط في نهاية النبضة رقم 3. و لم يحدث أي تغيير في خرج المرجاح بين ذلك الهبوط و الهبوط التالي في نهاية النبضة رقم 4. و في لحظة الهبوط في نهاية النبضة رقم 4 لم يحدث تغيير في خرج المرجاح لأن المرجاح ظل طوال تلك النبضة محتفظاً بآخر حالة وصل إليها.

مثال:

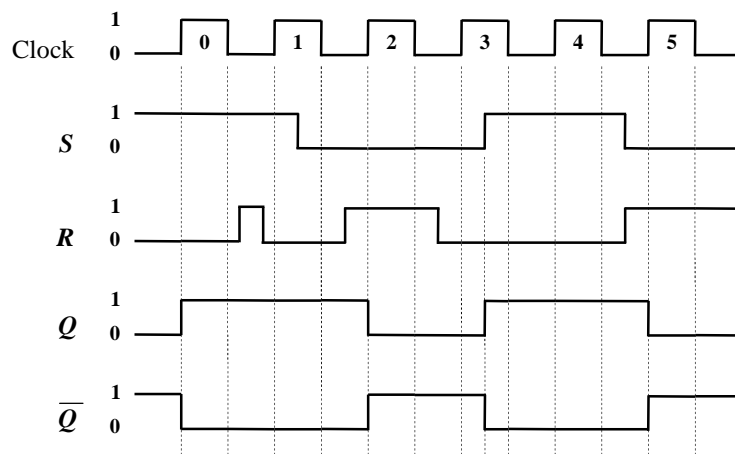
أكمل مخطط التزامن التالي و ذلك:

(أ) لمرجاح SR متزامن. (ب) لمرجاح القائد-التابع.

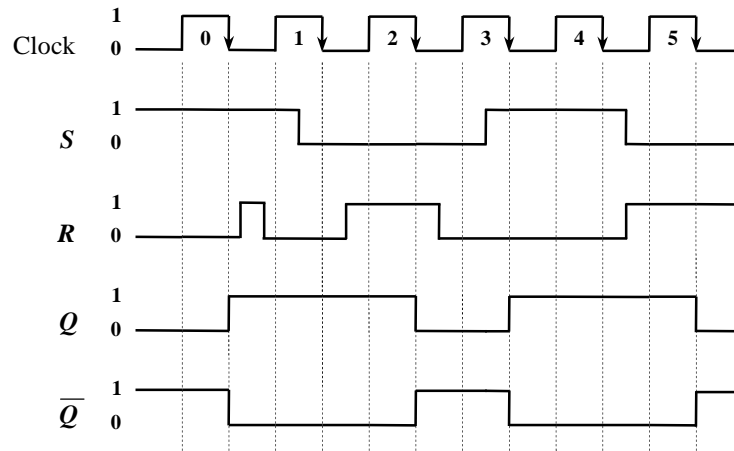


الحل:

(أ) لمرجاح SR متزامن

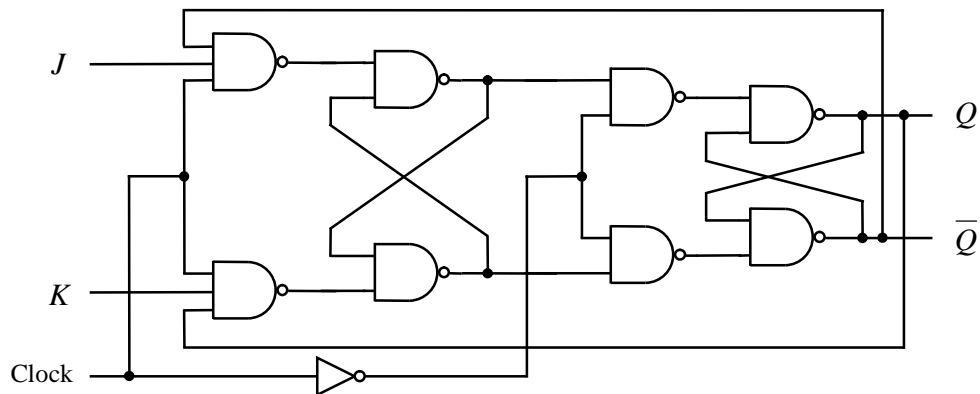


(ب) مرجاح القائد-التابع

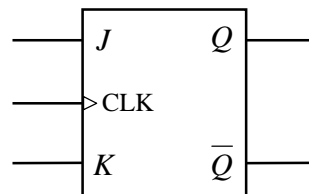


مرجاح JK (JK Flip Flop)

مرجاح JK هو عبارة عن مرجاح من نوع القائد-التابع مزود بتغذية مرتدة (Feedback) إضافية، كما هو موضح بالشكل التالي



و في ما يلي المخطط المنطقي و جدول الصواب لمرجاح JK

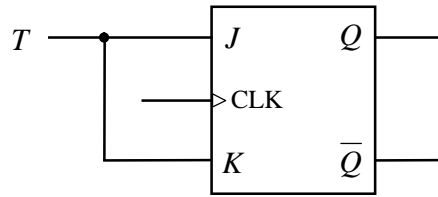


| C | J | K | Q_{n+1} | |
|-----|-----|-----|------------------|--------|
| 0 | × | × | Q_n | Keep |
| 1 | 0 | 0 | Q_n | Keep |
| 1 | 0 | 1 | 0 | RESET |
| 1 | 1 | 0 | 1 | SET |
| 1 | 1 | 1 | $\overline{Q_n}$ | Toggle |

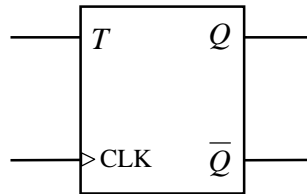
نلاحظ هنا أن جدول الصواب لمرجاح JK يشبه إلى حد كبير جدول الصواب لمرجاح القائد-التابع، حيث يحل الطرف J محل الطرف S في إجراء عملية SET للمرجاح و يحل الطرف K محل الطرف R في إجراء عملية RESET للمرجاح. و لكن يتميز مرجاح JK عن مرجاح القائد-التابع في عدم وجود دخل غير مسموح به أو غير مستخدم، حيث أن الدخل $J = 1$ و $K = 1$ يؤدي إلى عكس حالة المرجاح، و هي العملية التي تسمى Toggle.

مرجاح T (T Flip Flop)

و T هنا هي إختصار لكلمة Toggle، بمعنى عكس الحالة، كما سبق و أن أوضحنا. و مرجاح T هو عبارة عن مرجاح JK تم ربط طرفي الدخل له في طرف واحد هو الطرف T ، كما هو موضح بالشكل التالي



و في ما يلي المخطط المنطقي و جدول الصواب لمرجاح T



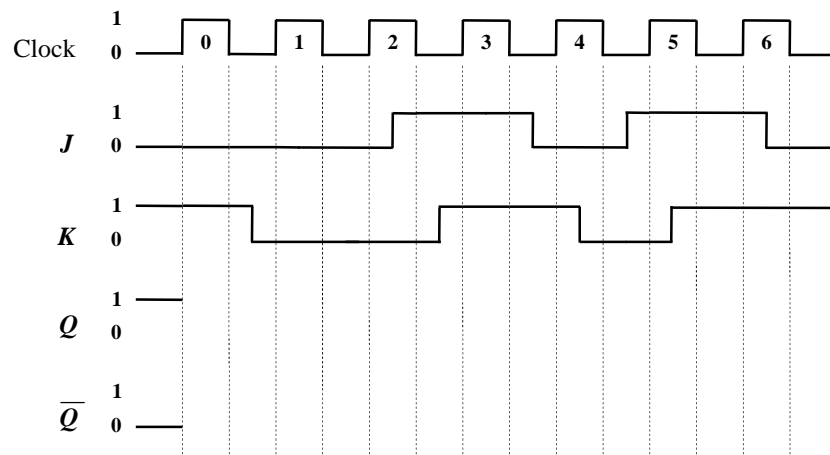
| C | T | Q_{n+1} | |
|-----|-----|------------------|--------|
| 0 | × | Q_n | Keep |
| 1 | 0 | Q_n | Keep |
| 1 | 1 | $\overline{Q_n}$ | Toggle |

لاحظ عدم إمكانية إجراء عملية SET أو عملية RESET لمراجيح T، بل يمكن فقط الاحتفاظ بحالته السابقة أو عكس تلك الحالة.

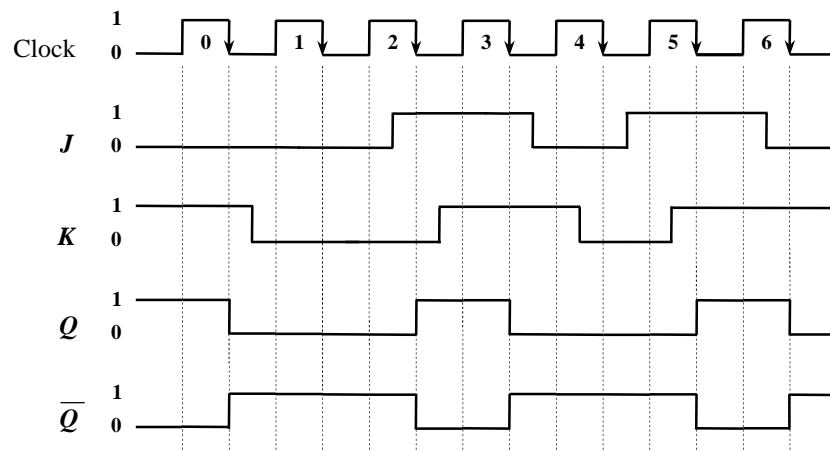
تستخدم مراجيح JK و مراجيح T في بناء العدّادات (Counters).

مثال:

أكمل مخطط التزامن التالي لمراجيح JK

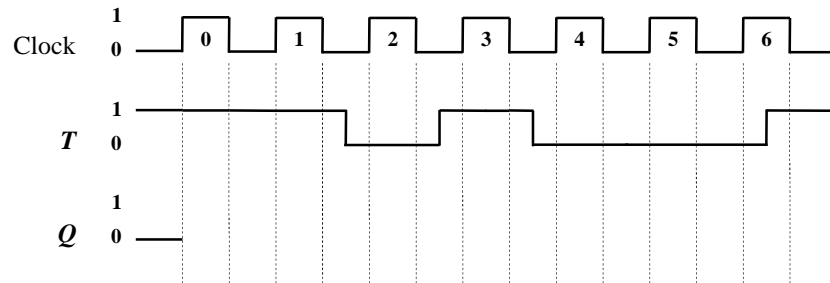


الحل:

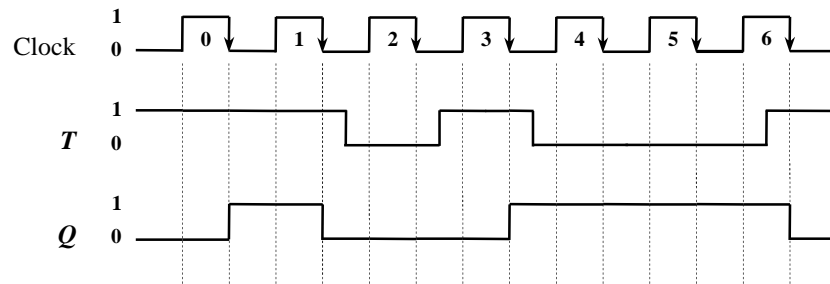


مثال:

أكمل مخطط التزامن التالي لمرجاح T

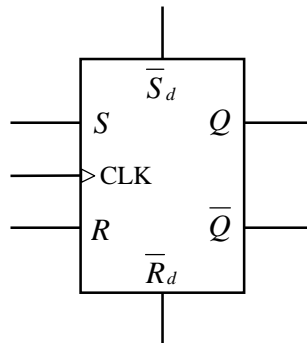


الحل:



4-2 أطراف الدخول المباشر (Direct Inputs)

في بعض الأحيان قد يكون مطلوباً تغيير حالة المرحاح بصورة استثنائية، بغض النظر عن حالة إشارة التزامن. فعلى سبيل المثال قد يكون مطلوباً وضع حالة ابتدائية في المرحاح في غياب إشارة التزامن، بحيث يبدأ المرحاح العمل من تلك الحالة عندما تبدأ نبضات التزامن، أو قد يكون مطلوباً تغيير التسلسل الطبيعي الذي تمر به حالات المرحاح و وضع حالة معينة فيه بصورة استثنائية. تستخدم لهذا الغرض أطراف الدخول المباشر (Direct Inputs)، التي تسمى أيضاً بأطراف الدخول غير المتزامن (Asynchronous Inputs)، والتي يرمز لها بـ \bar{S}_d و \bar{R}_d . والشكل التالي يمثل مرجاحاً من نوع القائد-التابع مزود بأطراف دخل مباشر



حيث يستخدم الطرف \bar{S}_d في إجراء عملية SET للمرجاح بصورة مباشرة، و يستخدم الطرف \bar{R}_d في إجراء عملية RESET للمرجاح بصورة مباشرة. أحياناً يرمز لأطراف الدخل المباشر بـ PRESET و CLEAR.

لاحظ أن أطراف الدخل المباشر نشطة منخفضة (Active Low)، بمعنى أن العملية المطلوبة يتم إجراؤها بوضع 0 في الطرف المقابل لها. فإجراء عملية SET بصورة مباشرة نضع القيمة 0 على الطرف \bar{S}_d و القيمة 1 على الطرف \bar{R}_d ، وإجراء عملية RESET بصورة مباشرة نضع القيمة 0 على الطرف \bar{R}_d و القيمة 1 على الطرف \bar{S}_d ، و العملية المطلوبة هنا تتم بغض النظر عن حالة إشارة التزامن. أما إذا تم وضع القيمة 1 على كلا الطرفين \bar{S}_d و \bar{R}_d فإن المرجاح يستجيب لأطراف الدخل المتزامن (S و R و CLK) بالصورة المعتادة. أما الدخل $\bar{S}_d = 0$ و $\bar{R}_d = 0$ فهو دخل غير مسموح به لأنه يجعل كلا طرفي الخرج Q و \bar{Q} مساويين 1. و يمكن تلخيص ذلك في جدول الصواب التالي

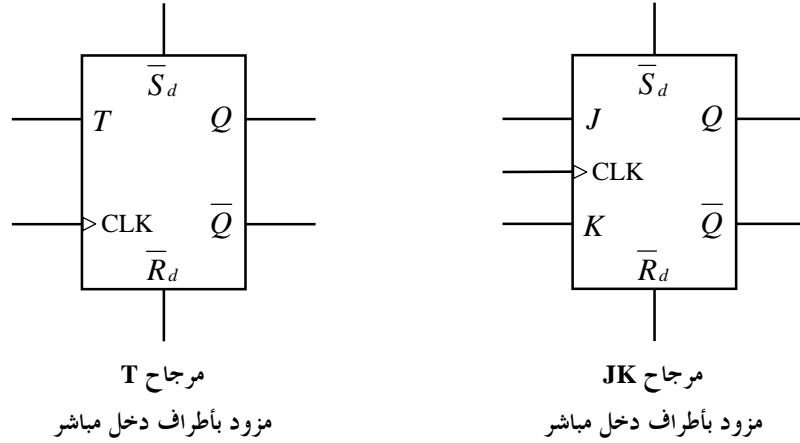
| \bar{S}_d | \bar{R}_d | Q_{n+1} | |
|-------------|-------------|------------------------|--------------|
| 0 | 0 | Not Used | |
| 0 | 1 | 1 | SET Direct |
| 1 | 0 | 0 | RESET Direct |
| 1 | 1 | استجابة للدخل المتزامن | |

لاحظ أن أي طرف من أطراف الدخل المباشر مطلوب وضع القيمة المنطقية 1 فيه يمكن تركه دون توصيل، أي تركه مفتوحاً (Open). فمثلاً لإجراء عملية SET نضع 0 على الطرف \bar{S}_d و نترك الطرف \bar{R}_d بدون توصيل، و لجعل المرجاح يستجيب للدخل المتزامن نترك كلا الطرفين \bar{S}_d و \bar{R}_d بدون توصيل. و السبب في ذلك هو أنه في الدوائر المنطقية المنتمية لعائلة TTL (Transistor Transistor Logic)، و هي من أكثر عائلات المنطق شيوعاً في الاستخدام، ترك الطرف دون توصيل يكافئ وضع القيمة المنطقية 1 فيه.

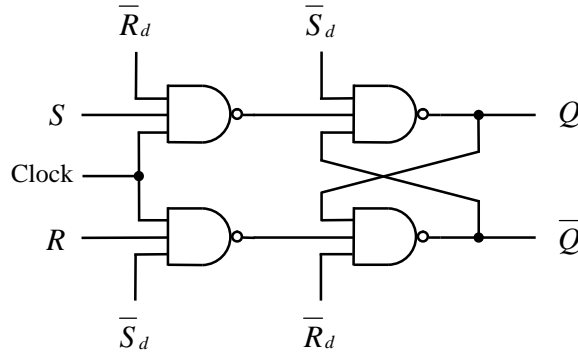
و في ما يلي جدول الصواب لمرجاح SR متزامن (أو مرجاح القائد-التابع) مزود بأطراف دخل مباشر

| \bar{S}_d | \bar{R}_d | C | S | R | Q_{n+1} | \bar{Q}_{n+1} | |
|-------------|-------------|---|---|---|-----------|-----------------|--------------|
| 0 | 0 | × | × | × | 1 | 1 | Invalid |
| 0 | 1 | × | × | × | 1 | 0 | SET Direct |
| 1 | 0 | × | × | × | 0 | 1 | RESET Direct |
| 1 | 1 | 0 | × | × | Q_n | \bar{Q}_n | Keep |
| 1 | 1 | 1 | 0 | 0 | Q_n | \bar{Q}_n | Keep |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 | RESET |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 | SET |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | Invalid |

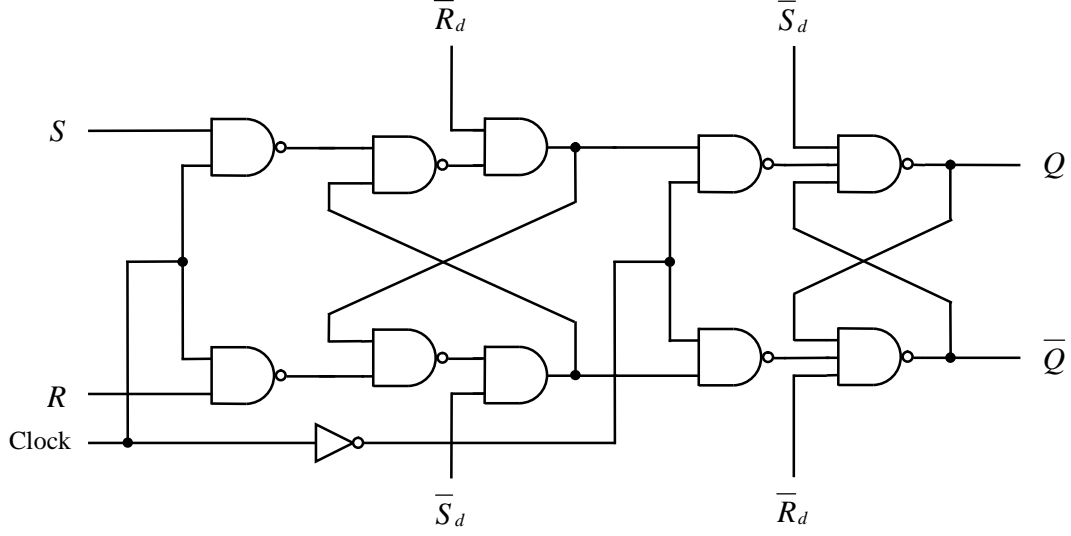
أي نوع من أنواع المراجع المتزامنة التي درسناها يمكن أن يكون مزوداً بأطراف دخل مباشر (Direct Inputs)، و يرمز لأطراف الدخل المباشر في هذه الحالات دائماً بـ \bar{S}_d و \bar{R}_d بغض النظر عن نوع المرجح، كما هو موضح أدناه



الشكل التالي يوضح كيفية ظهور أطراف الدخل المباشر في الدائرة المنطقية لمرجح SR المتزامن



الشكل التالي يوضح كيفية ظهور أطراف الدخل المباشر في الدائرة المنطقية لمرجاح القائد-التابع

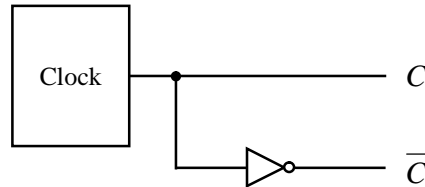


تدريب 1:

ارسم الدائرة المنطقية لمرجاح JK مزود بأطراف دخل مباشر.

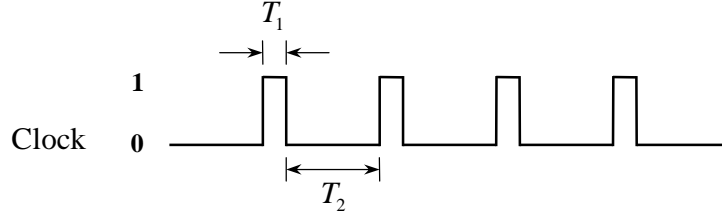
5-2 التزامن ثنائي الطور (Two-Phase Clocking)

في معظم الأنظمة الرقمية تستخدم مراجيح من نوع القائد-التابع (Master-Slave Flip Flops). وفي هذا النوع من المراجع، كما نعلم، تدخل إشارة التزامن (Clock) مباشرة إلى مرحلة القائد (Master)، وتدخل معكوسة إلى مرحلة التابع (Slave)، ويتم عكس إشارة التزامن باستخدام عاكس منطقي موجود داخل كل مرجاح. ويمكن الإستغناء عن كل هذه العواكس المنطقية إذا قمنا بعكس إشارة التزامن عند مصدرها باستخدام عاكس منطقي واحد، كما هو موضح أدناه

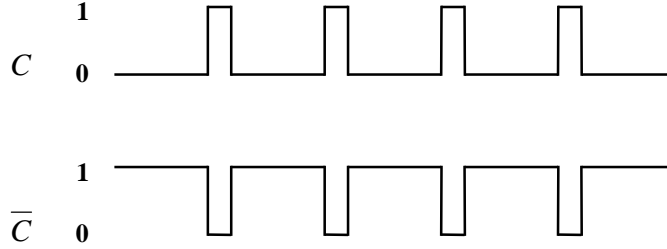


فنجصل على إشارتين؛ إشارة التزامن C و معكوسها \bar{C} . يتم بعد ذلك توزيع الإشارتين على أجزاء النظام الرقمي بحيث تدخل الإشارة C إلى مرحلة القائد (Master) و الإشارة \bar{C} إلى مرحلة التابع (Slave) من كل مرجاح.

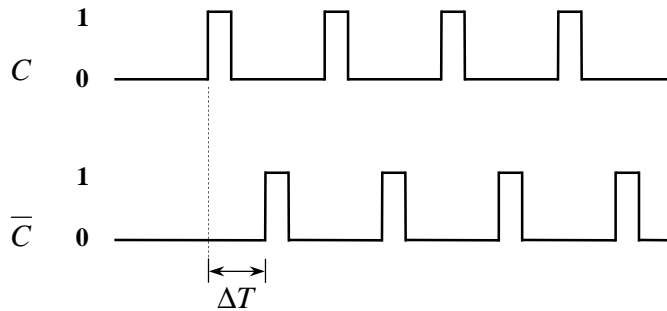
و لكن تظهر هنا مشكلة ناتجة عن شكل إشارة التزامن (Clock)، و الموضح أدناه



نلاحظ أن إشارة التزامن (Clock) عبارة عن سلسلة من النبضات الضيقة تفصل بينها فترات إنتظار واسعة نسبياً، أي أن الفترة الزمنية T_1 التي تكون فيها إشارة التزامن مرتفعة (High) أقصر بكثير من الفترة الزمنية T_2 التي تكون فيها الإشارة منخفضة (Low). فإذا ما قمنا بعكس هذه الإشارة باستخدام عاكس منطقي نحصل على النتيجة الموضحة بالشكل التالي



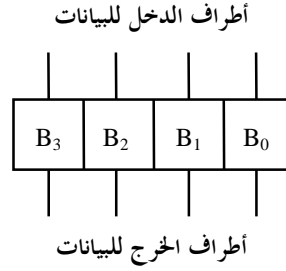
نلاحظ أن الإشارة \bar{C} هنا مكونة من نبضات واسعة و فترات إنتظار ضيقة، و مثل هذه الإشارة قد لا تصلح كإشارة تزامن نظراً إلى أن فترات الإنتظار الضيقة قد لا تكون كافية لتلاشي الـ Hazards. لذلك لا يتم عادة الحصول على الإشارة \bar{C} بعكس الإشارة C باستخدام عاكس منطقي و إنما يتم الحصول عليها بعمل إزاحة زمنية (Time Shift) للإشارة C ، كما هو موضح أدناه



تسمى الإشارة C بالطور الأول (Phase 1) من إشارة التزامن، و يرمز لها بالرمز f_1 ، و تسمى الإشارة \bar{C} بالطور الثاني (Phase 2) من إشارة التزامن، و يرمز لها بالرمز f_2 . و يطلق على هذا الأسلوب في التزامن تسمية التزامن ثنائي الطور (Two-Phase Clocking).

3- المسجلات (Registers)

المسجل (Register) هو عبارة عن موقع تخزيني له القدرة على اختزان معلومة مكونة من عدة خانات. و الشكل التالي يوضح المخطط المنطقي لمسجل مكون من أربعة خانات (4-bit Register)

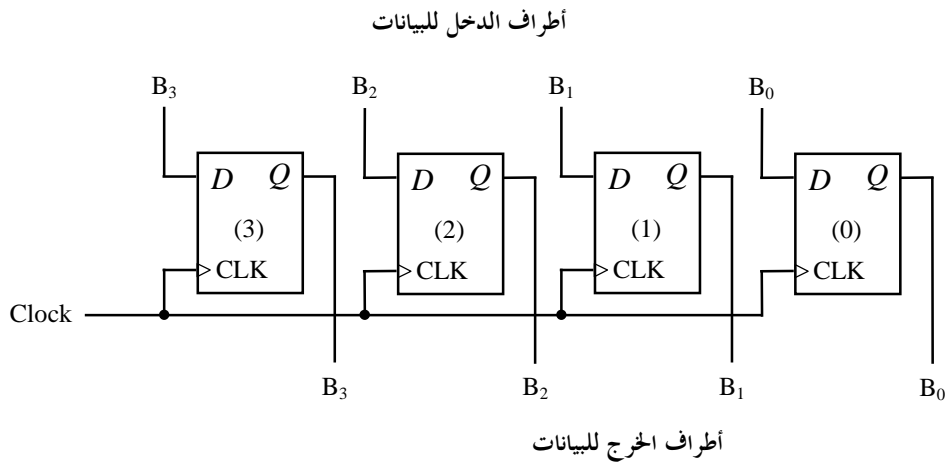


و العمليات التي يمكن إجراؤها على المسجلات هي:

1. الكتابة (Write)، أي تخزين معلومة في المسجل.
2. القراءة (Read)، أي إسترجاع معلومة مخزنة في المسجل.
3. نقل البيانات ما بين المسجلات (Register-to-Register Transfer).

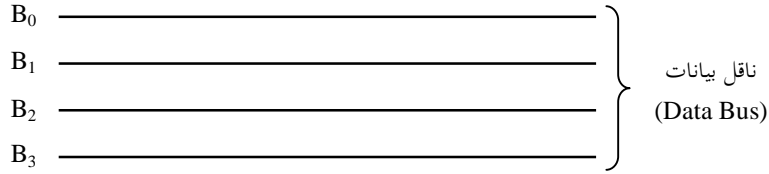
3-1 بناء المسجلات

يتم بناء المسجلات باستخدام مراجيح D (D Flip Flops)، و نحتاج عدداً من المراجع بعدد الخانات الثنائية (bits) المطلوب تخزينها. الشكل التالي يوضح الدائرة المنطقية لمسجل مكون من أربعة خانات (4-bit Register)



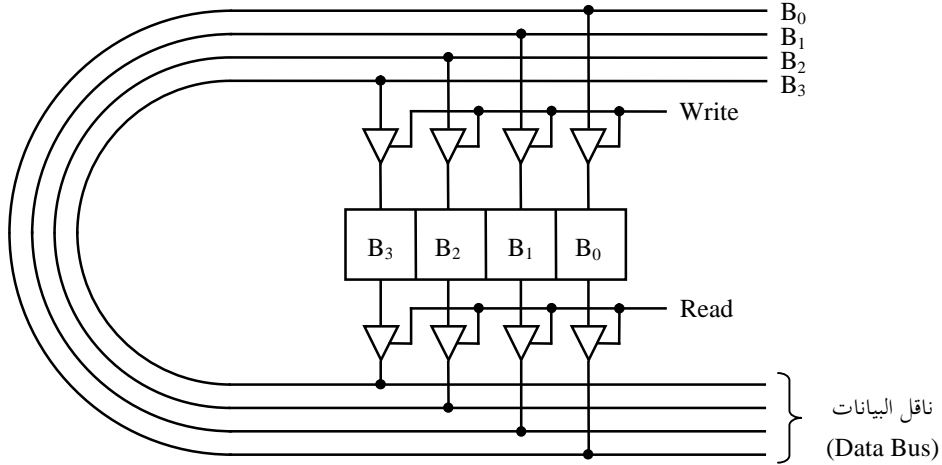
2-3 الكتابة في المسجلات و القراءة منها (Write and Read Operations)

عند إجراء عملية كتابة (Write) في المسجل فإن المعلومة المطلوب تخزينها عادة ما تصل إلى المسجل من خلال ناقل بيانات (Data Bus)، و عند إجراء عملية قراءة (Read) من المسجل فإن المعلومة التي تم إسترجاعها عادة ما تنقل من المسجل إلى الجهة المقصودة عبر ناقل البيانات (Data Bus) أيضاً. و ناقل البيانات هذا هو عبارة عن مجموعة من الموصلات المتوازية كل منها يحمل bit واحد فقط من البيانات، و الشكل التالي يوضح ناقل بيانات ذو أربعة خانات (4-bit Data Bus)

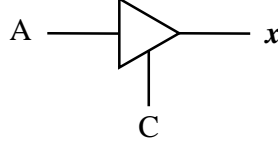


لاحظ أنه من الناحية الكهربائية لا بد من وجود موصل خامس في ناقل البيانات ذو الأربعة خانات الموضح أعلاه، و هذا الموصل الخامس هو الموصل الأرضي (Ground أو GND) الذي يعتبر مرجع قياس الجهود بالنسبة لبقية الموصلات. و لكن لا يتم عادة توضيح هذا الموصل الأرضي و إنما يُفهم وجوده ضمناً، و ذلك كنوع من التبسيط.

هذا و يتم ربط كل من أطراف الدخل للبيانات و أطراف الخرج للبيانات للمسجل بناقل البيانات باستخدام عوازل ثلاثية الحالة (Tristate Buffers)، كما هو موضح أدناه



و العازل ثلاثي الحالة (Tristate Buffer) هو عبارة عن بوابة منطقية لها طرف دخل A و طرف خرج x و طرف تحكم C، كما هو موضح أدناه



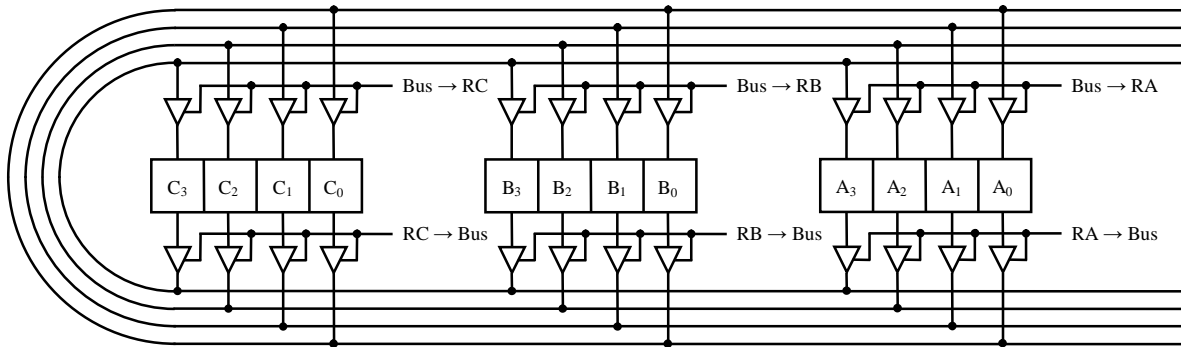
عند وضع القيمة 1 في طرف التحكم C يمر الدخل كما هو إلى الخرج، أي يكون $x = A$ ، أما عند وضع القيمة 0 في طرف التحكم C تدخل البوابة في الحالة الثالثة أي حالة المعاوقة العالية (High Impedance)، و فيها يتم عزل خرج البوابة عن دخلها بمعاوقة عالية.

- لإجراء عملية كتابة (Write) للبيانات الظاهرة على الناقل (Bus) في المسجل نقوم بجعل الإشارة Write مساوية 1، فيتم توصيل أطراف الدخل للمسجل مع الناقل، و تنتقل البيانات الموجودة على الناقل إلى داخل المسجل و يتم إختزائها. بعد ذلك يجب إعادة الإشارة Write إلى 0 مرة أخرى لفصل أطراف الدخل للمسجل عن الناقل، و ذلك لإخلاء الناقل بحيث يكون متاحاً للاستخدام في عمليات نقل بيانات أخرى.

- لإجراء عملية قراءة (Read) للبيانات المخزنة في المسجل نقوم بجعل الإشارة Read مساوية 1، فيتم توصيل أطراف الخرج للمسجل مع الناقل، و تظهر البيانات المخزنة في المسجل على الناقل و تكون متاحة لقراءتها من الناقل بواسطة أي جهة طالبة لها. بعد ذلك يجب إعادة الإشارة Read إلى 0 مرة أخرى لفصل أطراف الخرج للمسجل عن الناقل، و ذلك لإخلاء الناقل بحيث يكون متاحاً للاستخدام في عمليات نقل بيانات أخرى.

3-3 نقل البيانات بين المسجلات (Register-to-Register Transfer)

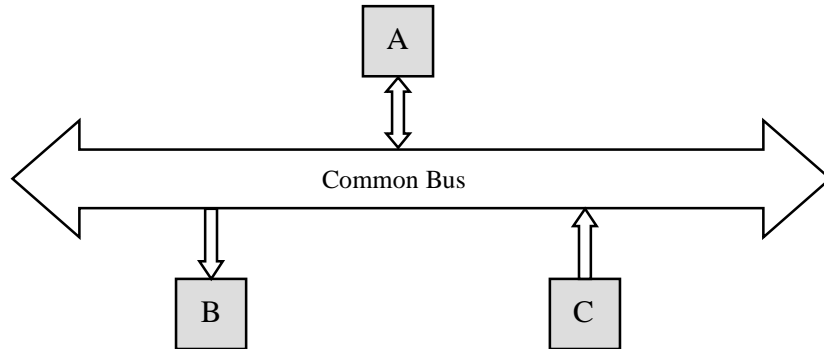
لنقل البيانات بين مجموعة من المسجلات يتم ربط تلك المسجلات بناقل مشترك (Common Bus)، كما هو موضح أدناه



لنقل البيانات من مسجل إلى آخر يتم إستخدام الناقل (Bus) كوسيط، حيث يتم قراءة محتويات المسجل الأول لتظهر تلك المحتويات على الناقل، بعد ذلك يتم قراءتها من الناقل بواسطة المسجل الثاني. مثلاً

- لإجراء عملية نقل البيانات $RA \rightarrow RB$ (أي نسخ محتويات المسجل RA للمسجل RB).
 1. نجعل الإشارة $RA \rightarrow Bus$ مساوية 1 فتظهر محتويات المسجل RA على الناقل.
 2. نجعل الإشارة $Bus \rightarrow RB$ مساوية 1 فتنقل البيانات الظاهرة على الناقل إلى المسجل RB .
 3. نعيد الإشارتين $RA \rightarrow Bus$ و $Bus \rightarrow RB$ إلى 0 مرة أخرى لإخلاء الناقل.
- لإجراء عملية نقل البيانات $RA \rightarrow \begin{cases} RA \\ RC \end{cases}$ (أي نسخ محتويات المسجل RA للمسجلين RC و RA).
 1. نجعل الإشارة $RA \rightarrow Bus$ مساوية 1 فتظهر محتويات المسجل RA على الناقل.
 2. نجعل الإشارتين $Bus \rightarrow RA$ و $Bus \rightarrow RC$ مساويتين 1 فتنقل البيانات الظاهرة على الناقل إلى كلا المسجلين RA و RC .
 3. نعيد الإشارات $RA \rightarrow Bus$ و $Bus \rightarrow RA$ و $Bus \rightarrow RC$ إلى 0 مرة لإخلاء الناقل.

هذا و من الشائع في الأنظمة الرقمية إستخدام ناقل مشترك (Common Bus) لنقل البيانات بين الأجزاء المختلفة للنظام الرقمي، كما هو موضح أدناه



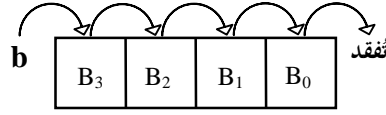
هذا و يعتبر عرض الناقل، أي عدد الـ bits التي يحملها، من العوامل الهامة جداً في تحديد سرعة عمل النظام الرقمي، فكلما زاد عرض الناقل أمكن نقل كمية أكبر من البيانات عبره في عملية النقل الواحدة.

4-3 مسجلات الإزاحة (Shift Registers)

مسجل الإزاحة (Shift Register) هو عبارة عن مسجل يستطيع، إضافة إلى العمليات السابقة، عمل إزاحة للبيانات الموجودة بداخله بمقدار خانة واحدة أو أكثر يميناً أو يساراً. وهناك عدة أنواع من الإزاحة

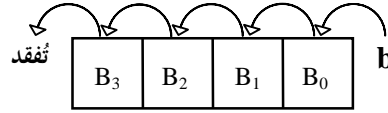
- الإزاحة إلى اليمين (Shift Right):

هنا تتم الإزاحة بمقدار خانة واحدة إلى اليمين حيث تُفقد الخانة الدنيا B_0 و تحل الخانة B_1 محلها، و تحل الخانة B_2 محل الخانة B_1 ، و تحل الخانة B_3 محل الخانة B_2 ، و يتم إدخال bit من الخارج إلى الخانة العليا (MSB) ليحل محل الخانة B_3 . كما هو موضح أدناه



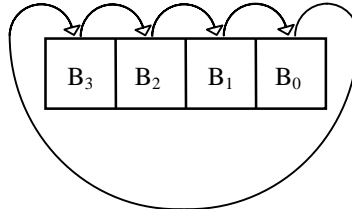
- الإزاحة إلى اليسار (Shift Left):

هنا تتم الإزاحة بمقدار خانة واحدة إلى اليسار حيث تُفقد الخانة العليا B_3 و تحل الخانة B_2 محلها، و تحل الخانة B_1 محل الخانة B_2 ، و تحل الخانة B_0 محل الخانة B_1 ، و يتم إدخال bit من الخارج إلى الخانة الدنيا (LSB) ليحل محل الخانة B_0 . كما هو موضح أدناه



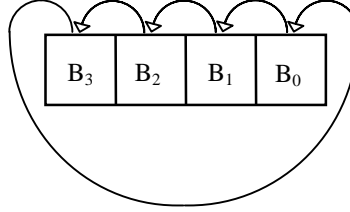
- الإزاحة الدورانية إلى اليمين (Rotate Right):

هنا تتم الإزاحة بمقدار خانة واحدة إلى اليمين و لكن لا يحدث أي فقد أو إدخال من الخارج، حيث أن الخانة الدنيا B_0 تحل محل الخانة العليا B_3 . كما هو موضح أدناه



- الإزاحة الدورانية إلى اليسار (Rotate Left):

هنا تتم الإزاحة بمقدار خانة واحدة إلى اليسار بدون أي فقد أو إدخال من الخارج، حيث أن الخانة العليا B_3 تحل محل الخانة الدنيا B_0 . كما هو موضح أدناه

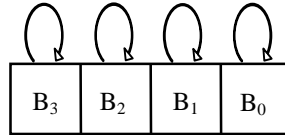


لاحظ أن كل أنواع الإزاحة الموضحة أعلاه تحدث بصورة متزامنة، أي مرتبطة بإشارة التزامن (Clock)، فمع كل نبضة من نبضات التزامن تحدث إزاحة بمقدار خانة واحدة في الاتجاه المحدد، و تستمر الإزاحة ما دامت إشارة التزامن مستمرة.

و هناك عمليات أخرى، بخلاف عملية الإزاحة بأنواعها، يمكن إجراؤها على مسجلات الإزاحة مثل:

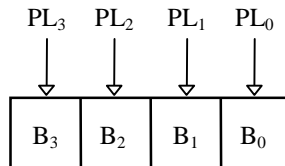
- التوقف (Hold):

المقصود هنا هو إيقاف عملية الإزاحة الجارية بصورة مؤقتة، و يمكن أن يتم ذلك بالطبع بإيقاف إشارة التزامن (Clock)، حيث أن الإزاحة مرتبطة بإشارة التزامن كما ذكرنا من قبل، و لكن الأسلوب الأفضل هنا هو أن يتم ذلك بأن تحل كل خانة من خانات المسجل محل نفسها، كما هو موضح أدناه



- التعبئة على التوازي (Parallel Load):

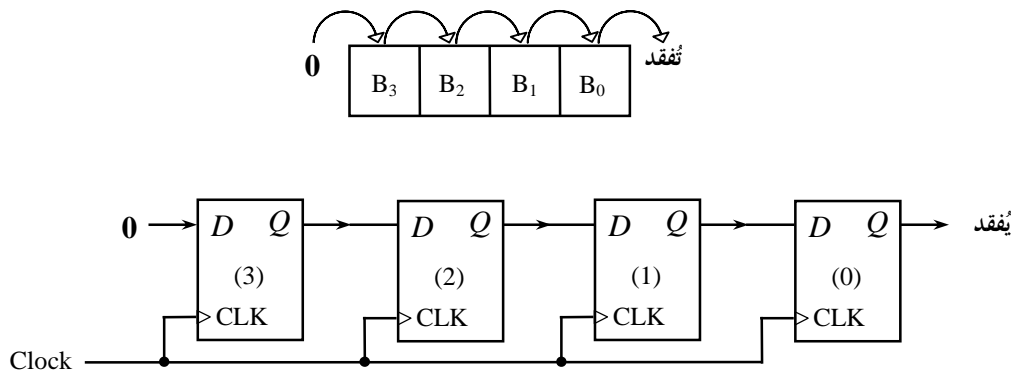
المقصود هنا هو تعبئة المسجل بالبيانات من الخارج إستعداداً للبدء بعملية الإزاحة، كما هو موضح أدناه



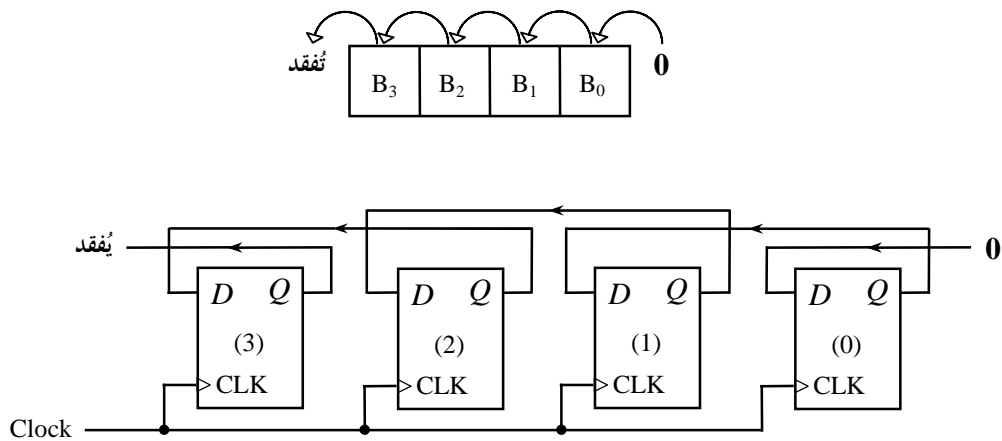
بناء مسجلات الإزاحة

كما ذكرنا من قبل فإن بناء المسجلات يتم باستخدام مراجيح D (D Flip Flops)، و نحتاج عدداً من المراجع بعدد الخانات الثنائية (bits) المكونة للمسجل. و في مسجلات الإزاحة يتم ربط المراجع مع بعضها البعض على التوالي بحيث يكون خرج كل مرجح دخلاً للمرجح المجاور له و ذلك حسب إتجاه الإزاحة المطلوب.

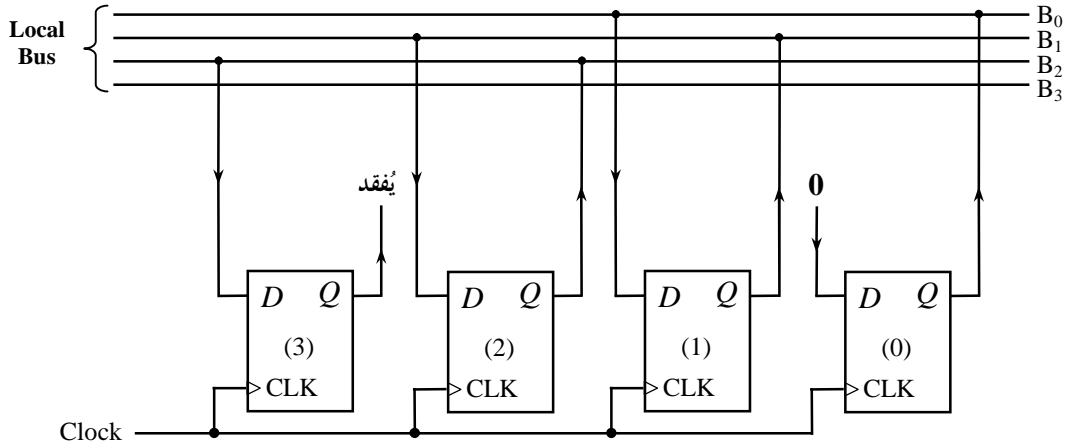
الشكل التالي يوضح المخطط المنطقي و الدائرة المنطقية لمسجل إزاحة إلى اليمين، مع الملء بأصفار، مكون من أربعة خانات (4-bit Shift Right Zero-Fill Register)



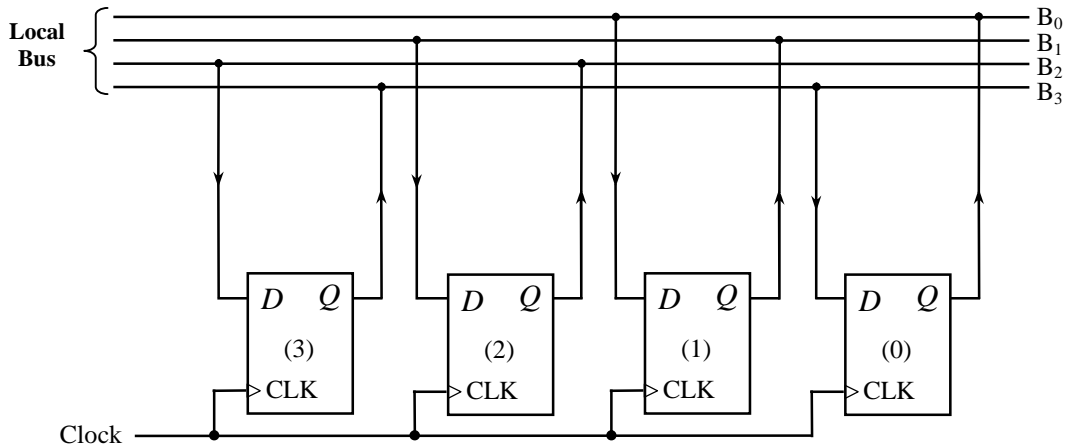
الشكل التالي يوضح المخطط المنطقي و الدائرة المنطقية لمسجل إزاحة إلى اليسار، مع الملء بأصفار، مكون من أربعة خانات (4-bit Shift-Left Zero-Fill Register)



نلاحظ أن الأسلوب المستخدم في رسم الدائرة أعلاه يجعل من الصعب متابعتها و فهم عملها، خصوصاً إذا كانت الدائرة أكبر و أكثر تعقيداً. لذلك سنستخدم في رسم دوائر مسجلات الإزاحة أسلوباً أفضل يجعل من السهل متابعتها و فهم عملها، بل و يسهل بنائها في المعمل أيضاً، و نستخدم في هذا الأسلوب ناقلاً محلياً (Local Bus) في ربط المراجيح المكونة للمسجل مع بعضها البعض. و نؤكد هنا على أن الناقل المستخدم هنا هو ناقل محلي (Local)، بمعنى أنه مضمن داخل دائرة مسجل الإزاحة، و يختلف عن الناقل المشترك (Common Bus) المستخدم في نقل البيانات بين أجزاء النظام الرقمي الذي أشرنا إليه من قبل. و الشكل التالي يوضح الدائرة المنطقية لمسجل الإزاحة إلى اليسار، مع الملء بأصفار، المكون من أربعة خانات (4-bit Shift-Left Zero-Fill Register) بعد إعادة رسمها بالأسلوب الجديد



و بتعديل طفيف على الدائرة الموضحة أعلاه يمكن تحويل المسجل إلى مسجل إزاحة دورانية إلى اليسار مكون من أربعة خانات (4-bit Rotate Left Register)، كما هو موضح بالشكل التالي

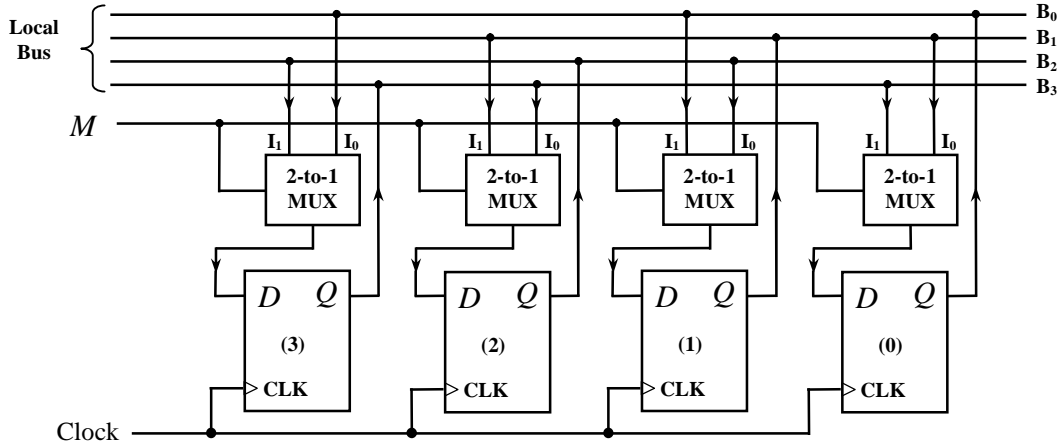


الإزاحة في الإتجاهين

المطلوب الآن تصميم مسجل إزاحة مكون من أربعة خانات و يستجيب لإشارة تحكم M ، بحيث يقوم بالإزاحة الدورانية إلى اليمين (Rotate Right) عندما تكون $M = 0$ ، و بالإزاحة الدورانية إلى اليسار (Rotate Left) عندما تكون $M = 1$.

نعلم أن الإزاحة إلى اليمين تتطلب ربط المراجيح مع بعضها البعض بطريقة معينة، و الإزاحة إلى اليسار تتطلب ربطها بطريقة أخرى مختلفة. فكيف يمكن ربط المراجيح بكلا الطريقتين في وقت واحد ثم إختيار إحدهما بناء على قيمة إشارة التحكم M ؟

يتم ذلك باستخدام دوائر دامج من نوع 1 إلى 2 (2-to-1 MUX's) كما هو موضح بالشكل التالي



استخدمنا هنا دوائر دامج من نوع 1 إلى 2 بعدد المراجيح المكونة للمسجل، حيث يحدد كل دامج دخل المرحاح المقابل له بناء على قيمة إشارة التحكم M التي تدخل إلى طرف الإختيار (Select Line) لكل دامج. فعندما تكون $M = 0$ يتم توصيل الطرف I_0 لكل دامج مع طرف الدخل للمرحاح المقابل، و يؤدي هذا لربط المراجيح بحيث تكون الإزاحة لليمين، و عندما تكون $M = 1$ يتم توصيل الطرف I_1 لكل دامج مع طرف الدخل للمرحاح المقابل، و يؤدي هذا لربط المراجيح بحيث تكون الإزاحة لليسار.

تدريب 2:

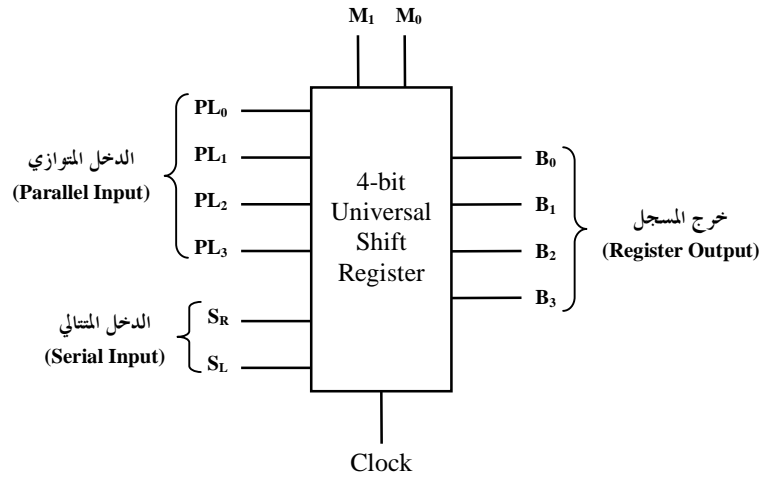
صمم مسجل إزاحة من أربعة خانات (4 bits) يستجيب لإشارة تحكم M ، فيقبل دخلاً متوازياً عندما تكون $M = 0$ ، و يقوم بالإزاحة إلى اليمين مع الملء بأصفار عندما تكون $M = 1$.

مسجل الإزاحة العام (Universal Shift Register)

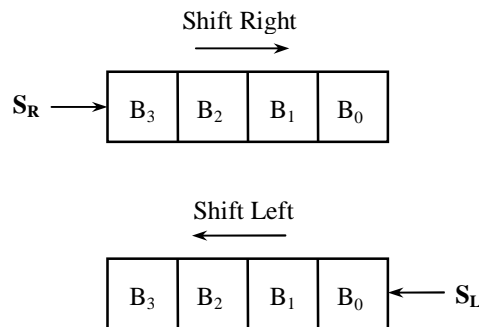
مسجل الإزاحة العام هو عبارة عن مسجل إزاحة يمكن أن يقبل دخلاً متوازياً (Parallel Input) أو دخلاً متتالياً (Serial Input)، و يقوم بالإزاحة يمينا أو يساراً أو يتوقف عن الإزاحة بناء على قيم إشارتي تحكم هما M_1 و M_0 ، كما هو موضح بالجدول التالي

| M_1 | M_0 | Operation |
|-------|-------|---------------|
| 0 | 0 | Hold |
| 0 | 1 | Shift Right |
| 1 | 0 | Shift Left |
| 1 | 1 | Parallel Load |

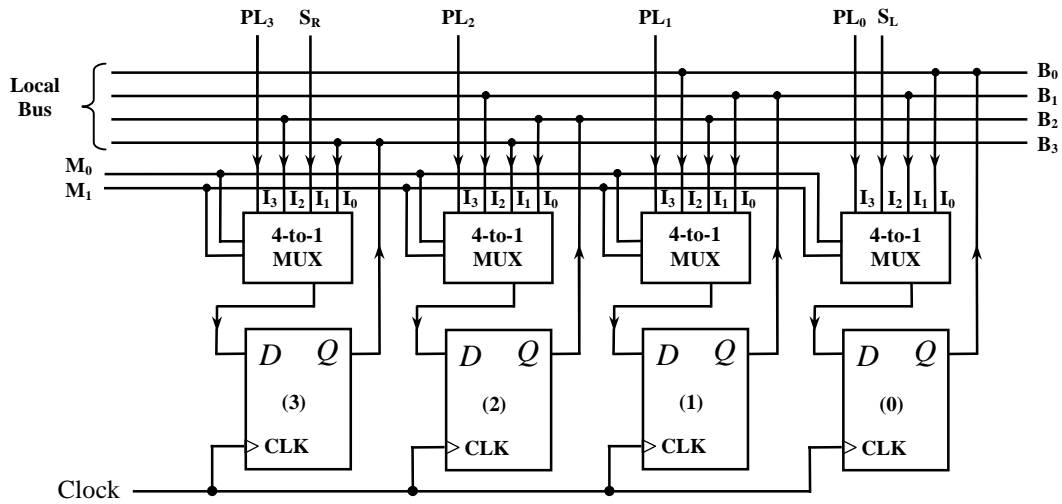
و الشكل التالي يوضح المخطط المنطقي لمسجل إزاحة عام ذو أربعة خانات (4-bit Universal Shift Register)



الدخل المتوازي (Parallel Input) يتم عبر أطراف الدخل المتوازي PL_0 ، PL_1 ، PL_2 ، و PL_3 . أما الدخل المتتالي (Serial Input) فيتم أثناء عملية الإزاحة إلى اليمين عبر طرف الدخل S_R إلى الخانة العليا (MSB) من المسجل، و أثناء عملية الإزاحة إلى اليسار عبر طرف الدخل S_L إلى الخانة الدنيا (LSB) من المسجل، كما هو موضح بالشكل التالي



نظراً إلى أن لدينا في هذا المسجل أربع طرق مختلفة لربط المراجع، يجب أن تكون جميعاً موجودة و أن يتم إختيار واحدة منها بناء على قيم إشارتي التحكم M_0 و M_1 ، فإنه يتم استخدام دوائر دامج من نوع 4 إلى 1 (4-to-1 MUX's) كما هو موضح بالشكل التالي الذي يمثل الدائرة المنطقية لمسجل إزاحة عام ذو أربعة خانات

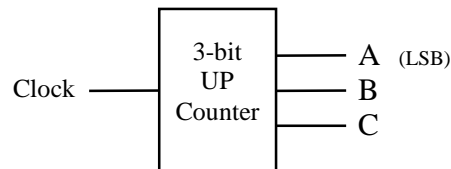


هذا و يمكن استخدام مسجل الإزاحة العام في تحويل البيانات من توازي إلى تنالي (Parallel to Serial) أو من تنالي إلى توازي (Serial to Parallel). فعند التحويل من توازي إلى تنالي يتم إدخال الـ bits للمسجل على التوازي بإجراء عملية تعبئة على التوازي (Parallel Load)، ثم يتم إجراء إزاحة إلى اليمين و الحصول على الـ bits على التتالي واحداً تلو الآخر بدءاً بالخانة الدنيا (LSB) عبر طرف الخرج B_0 ، أو إجراء إزاحة إلى اليسار و الحصول على الـ bits على التتالي واحداً تلو الآخر بدءاً بالخانة العليا (MSB) عبر طرف الخرج B_3 . و عند التحويل من تنالي إلى توازي يتم إدخال الـ bits إلى المسجل واحداً تلو الآخر بدءاً بالخانة الدنيا (LSB) عبر الطرف S_R أثناء عملية الإزاحة إلى اليمين، أو بدءاً بالخانة العليا (MSB) عبر الطرف S_L أثناء عملية الإزاحة إلى اليسار، و بعد إكمال دخول الـ bits يتم قراءتها على التوازي عبر أطراف الخرج للمسجل.

4- العدادات (Counters)

العداد (Counter) هو عبارة عن دائرة منطقية تتابعية لها القدرة على العد ثنائياً بترتيب معين. و ترتيب العد قد يكون ترتيباً تصاعدياً (Up Counting)، أو قد يكون ترتيباً تنازلياً (Down Counting)، أو قد يكون بأي ترتيب آخر. كل قيمة يصل إليها العداد أثناء عملية العد تسمى حالة (State)، و ينتقل العداد من حالة إلى أخرى من حالاته مع نبضات التزامن (Clock) و بترتيب معين. أي أن كل نبضة من نبضات التزامن تنقل العداد من الحالة التي هو فيها إلى الحالة التي تليها في ترتيب العد. و يمكن أن يبدأ العداد العد من أي حالة من حالاته، و يطلق على الحالة التي يبدأ العد منها تسمية الحالة الابتدائية (Initial State).

الشكل التالي يوضح المخطط المنطقي لعداد تصاعدي ذو ثلاثة خانات (3-bit Up Counter)

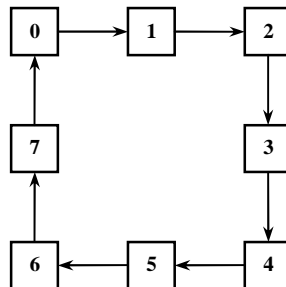


و الجدول التالي يوضح تسلسل العد (Counting Sequence) للعداد

| C | B | A | State |
|-------|---|---|-------|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 2 |
| 0 | 1 | 1 | 3 |
| 1 | 0 | 0 | 4 |
| 1 | 0 | 1 | 5 |
| 1 | 1 | 0 | 6 |
| 1 | 1 | 1 | 7 |
| <hr/> | | | |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 2 |
| M | M | M | M |

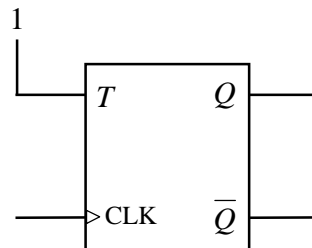
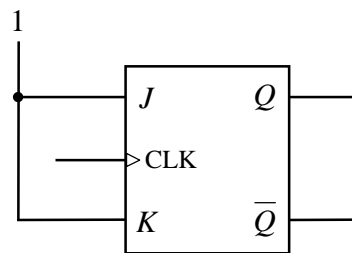
لاحظ أن للعداد ثمانية حالات هي 0، 1، 2، 3، 4، 5، 6، 7 يمر بها العداد بترتيب تصاعدي ابتداءً من الحالة 0 التي تعتبر الحالة الابتدائية (Initial State) للعداد. لاحظ أيضاً أن الخانة الدنيا (LSB) للعداد A تعكس حالتها مع كل نبضة من نبضات التزامن، في حين أن الخانة الثانية B تعكس حالتها كل نبضتين، و الخانة الثالثة C تعكس حالتها كل أربعة نبضات. لاحظ أيضاً أنه بعد وصول العداد إلى آخر حالة من حالاته في تسلسل العد يعود مرة أخرى إلى الحالة

الإبتدائية و يكرر العملية طالما كانت نبضات التزامن (Clock) مستمرة. هذا و يمكن توضيح حالات العدّاد و ترتيب المرور بها باستخدام مخطط يسمى بمخطط الحالات (State Diagram)، كما هو موضح أدناه

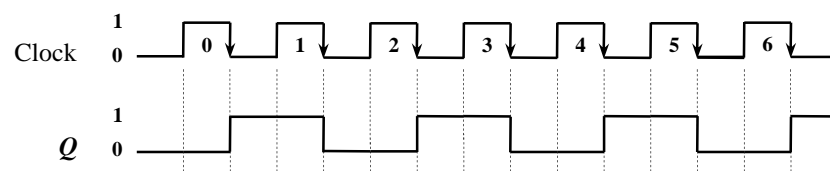


1-4 بناء العدّادات

يستخدم في بناء العدّادات مراجيح JK أو مراجيح T في وضع عكس الحالة (Toggle Mode)، كما هو موضح أدناه



و في هذا الوضع يقوم المرحاح بعكس حالته مع كل نبضة من نبضات التزامن، كما هو موضح أدناه



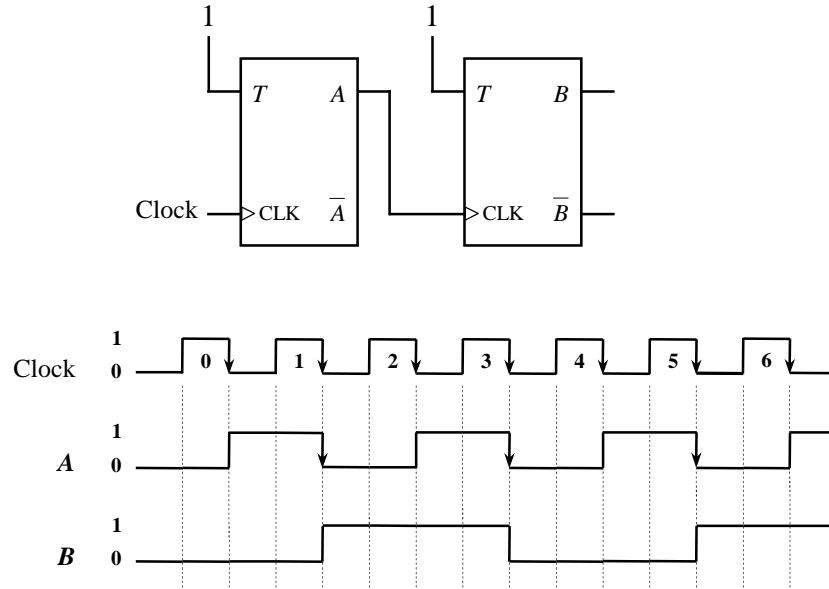
لاحظ أن الإشارة Q الخارجة من المرجاح يمكن اعتبارها أيضاً إشارة تزامن، و لكن إشارة التزامن الخارجة من المرجاح ترددها هو نصف تردد إشارة التزامن الداخلة إليه. أي أن المرجاح قد قام بقسمة تردد إشارة التزامن على 2، أو

$$f_Q = \frac{1}{2} f_C$$

حيث f_C هو تردد إشارة التزامن (Clock)، و f_Q هو تردد الإشارة Q الخارجة من المرجاح.

2-4 العد تصاعدياً (Up Counting)

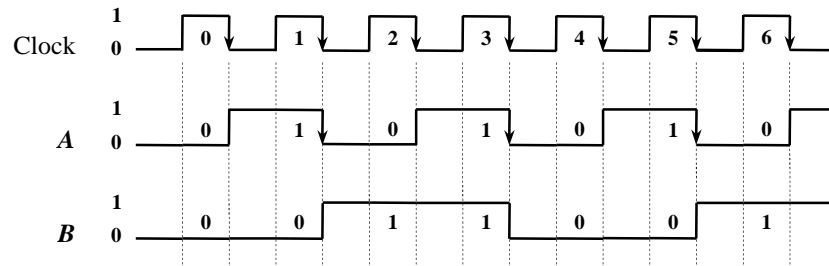
إذا قمنا بإدخال الإشارة الخارجة من المرجاح الأول كإشارة تزامن إلى مرجاح ثاني من نفس النوع فإن المرجاح الثاني سيقوم بقسمة تردد تلك الإشارة على 2 أيضاً، كما هو موضح أدناه



$$f_B = \frac{1}{2} f_A = \frac{1}{4} f_{Clock}$$

لاحظ أن المرجاح الأول A يعكس حالته مع كل نبضة من نبضات التزامن، و المرجاح الثاني B يعكس حالته كل نبضتين. و بناء عليه تصلح A أن تكون الخانة الدنيا (LSB)، و تصلح B أن تكون الخانة الثانية، في عدّاد تصاعدي ذو خانيتين (2-bit Up Counter)، أي أن الدائرة المنطقية أعلاه تمثل دائرة عدّاد تصاعدي ذو خانيتين.

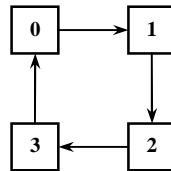
هذا و يمكن الحصول على تسلسل العد للعداد من مخطط التزامن (Timing Diagram) له كالتالي



و عليه يكون تسلسل العد (Counting Sequence) للعداد هو

| B | A | State |
|----------|----------|----------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 2 |
| 1 | 1 | 3 |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 2 |
| M | M | M |

و مخطط الحالات (State Diagram) للعداد هو

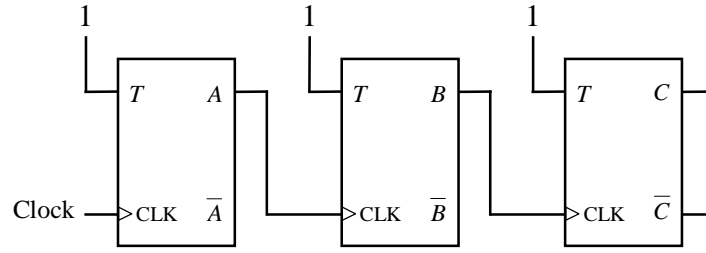


مثال:

صمم عدداً تصاعدياً ذو ثلاثة خانات (3-bit Up Counter) و أرسم مخطط التزامن له، ثم وضح تسلسل العد و مخطط الحالات، و ذلك إذا بدأ العداد العد من الحالة 3.

الحل:

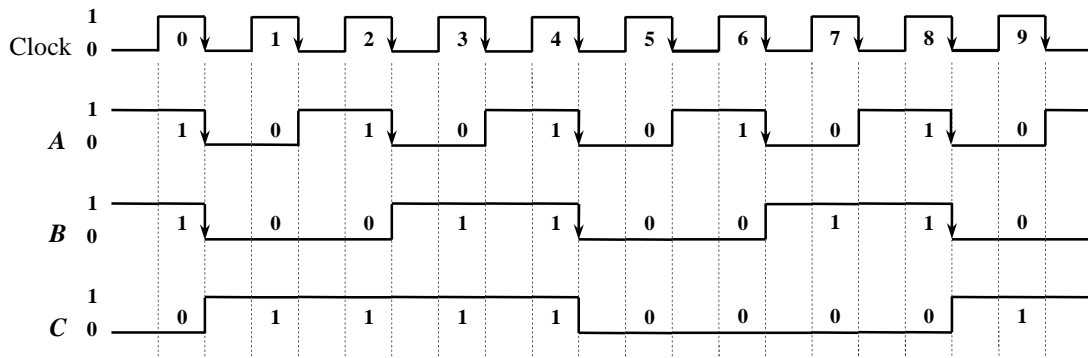
نحتاج عدداً من مراجيح T بعدد خانات العداد، أي ثلاثة مراجيح، و نقوم بإدخال الخرج غير المعكوس لكل مرجح كإشارة تزامن للمرجح الذي يليه، كما هو موضح أدناه



لرسم مخطط التزامن نحتاج لمعرفة الحالة الابتدائية لكل مرجاح من المراجيح الثلاثة، و يمكن معرفتها من الحالة الابتدائية للعداد كالتالي

$$C \ B \ A \\ 3 = (0 \ 1 \ 1)_2$$

أي أن الحالة الابتدائية لكل من A و B هي 1، و الحالة الابتدائية لـ C هي 0. و عليه يمكن رسم مخطط التزامن للعداد كالتالي

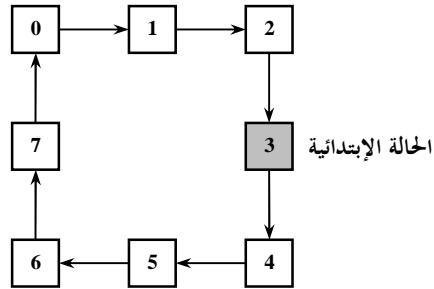


و من مخطط التزامن يمكن إيجاد تسلسل العد (Counting Sequence)

| C | B | A | State |
|---|---|---|-------|
| 0 | 1 | 1 | 3 |
| 1 | 0 | 0 | 4 |
| 1 | 0 | 1 | 5 |
| 1 | 1 | 0 | 6 |
| 1 | 1 | 1 | 7 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 2 |
| 0 | 1 | 1 | 3 |
| 1 | 0 | 0 | 4 |
| M | M | M | M |

لاحظ أنه رغم أن العدّاد قد بدأ العد من الحالة 3 إلا أن ترتيب العد تصاعدي كما هو مطلوب.

و مخطط الحالات (State Diagram) هو



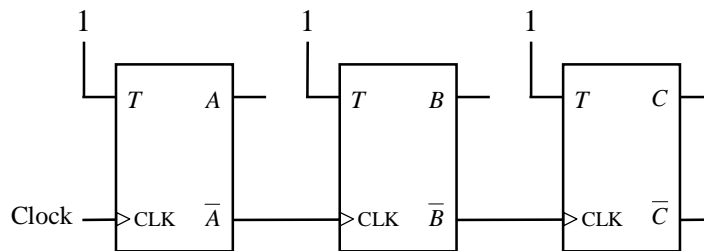
تدريب 3:

صمم عدّاداً تصاعدياً ذو أربعة خانات (4-bit Up Counter) و أرسم مخطط التزامن له، ثم وضع تسلسل العد و مخطط الحالات، و ذلك إذا بدأ العدّاد العد من الحالة 10.

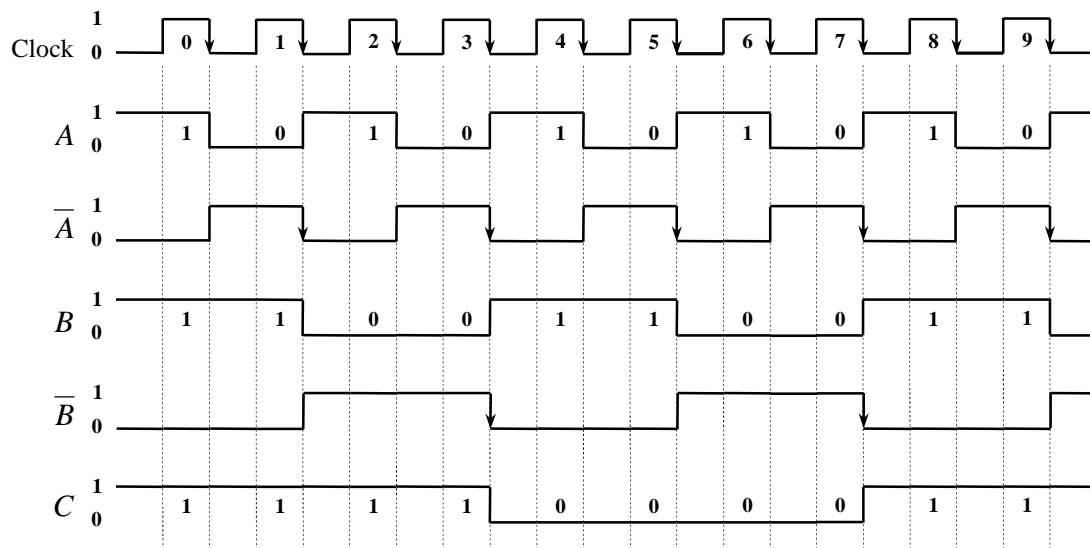
3-4 العد تنازلياً (Down Counting)

إذا استخدمنا الخرج المعكوس لكل مرجح كإشارة تزامن للمرجح الذي يليه فإن العدّاد الناتج يقوم بالعد تنازلياً.

الشكل التالي يوضح الدائرة المنطقية لعدّاد تنازلي مكون من ثلاثة خانات (3-bit Down Counter)



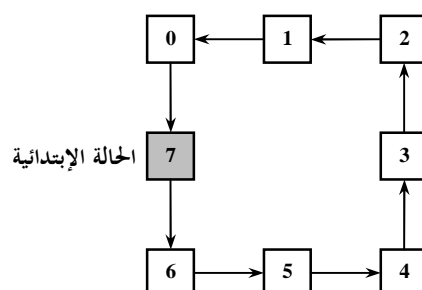
فإذا قمنا برسم مخطط التزامن للعدّاد مفترضين أنه بدأ العد من الحالة $(111)_2 = 7$ نحصل على



و من مخطط التزامن نجد أن تسلسل العد للعداد هو

| C | B | A | State |
|----------|----------|----------|----------|
| 1 | 1 | 1 | 7 |
| 1 | 1 | 0 | 6 |
| 1 | 0 | 1 | 5 |
| 1 | 0 | 0 | 4 |
| 0 | 1 | 1 | 3 |
| 0 | 1 | 0 | 2 |
| 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 7 |
| 1 | 1 | 0 | 6 |
| M | M | M | M |

أي أن العداد يقوم فعلاً بالعد تنازلياً، كما هو متوقع، ومخطط الحالات (State Diagram) له هو

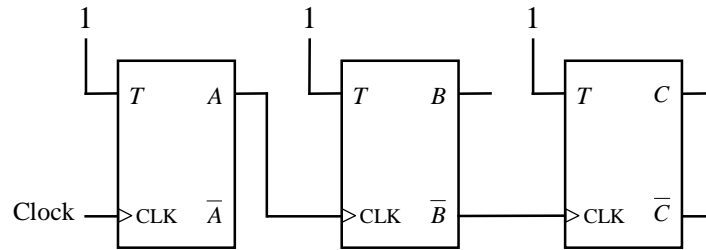


تدريب 4:

صمم عدّاداً تنازلياً ذو أربعة خانات (4-bit Down Counter) و أرسم مخطط التزامن له، ثم وضح تسلسل العد و مخطط الحالات، و ذلك إذا بدأ العدّاد العد من الحالة 12.

مثال:

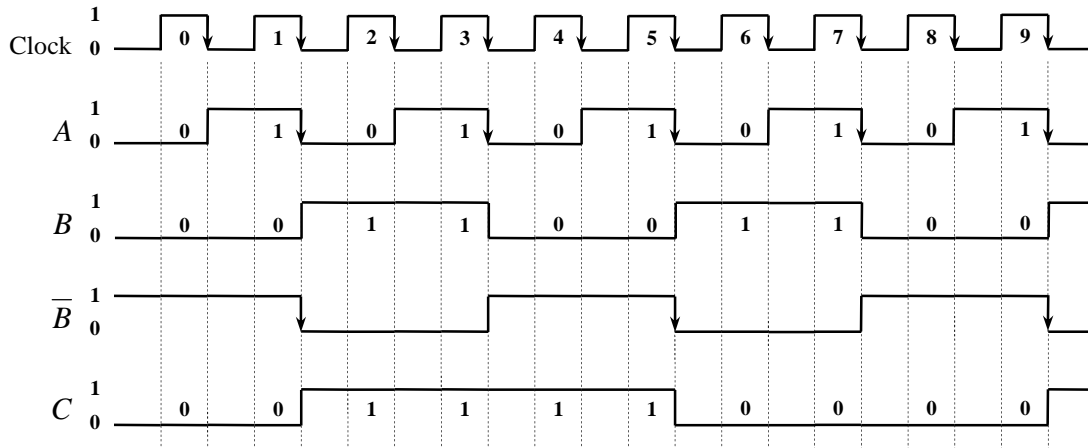
أرسم مخطط التزامن للعدّاد الموضحة دائرته المنطقية أدناه إذا بدأ العد من الحالة 0، ثم وضح تسلسل العد و أرسم مخطط الحالات له.



الحل:

من الواضح أن العدّاد هنا ليس عدّاداً تصاعدياً و لا هو عدّاد تنازلي.

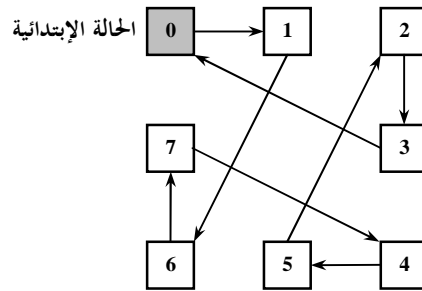
الحالة الابتدائية هنا هي $0 = (000)_2$



و من مخطط التزامن نجد أن تسلسل العد للعدّاد هو

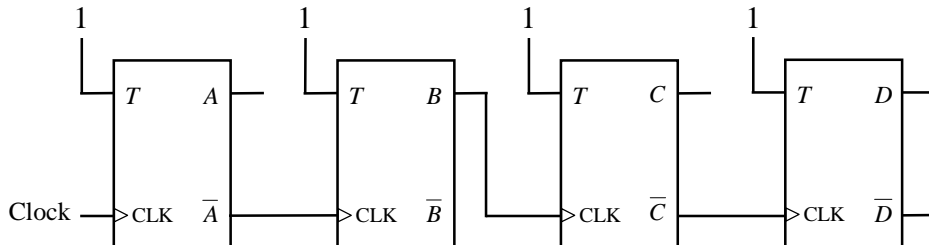
| C | B | A | State |
|---|---|---|-------|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 6 |
| 1 | 1 | 1 | 7 |
| 1 | 0 | 0 | 4 |
| 1 | 0 | 1 | 5 |
| 0 | 1 | 0 | 2 |
| 0 | 1 | 1 | 3 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| M | M | M | M |

ومخطط الحالات (State Diagram) للعدّاد هو



تدريب 5:

أرسم مخطط التزامن للعدّاد الموضحة دائرته المنطقية أدناه إذا بدأ العد من الحالة 5، ثم وضح تسلسل العد و أرسم مخطط الحالات له.



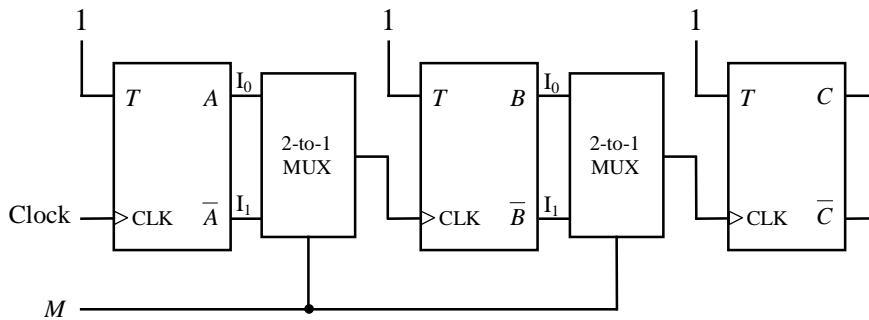
نلاحظ في جميع العدادات التي قمنا بدراستها حتى الآن أن إشارة التزامن الداخلة للمراجيح المختلفة المكونة للعداد ليست واحدة، لذلك يسمى هذا النوع من العدادات بالعدادات غير المتزامنة (Asynchronous Counters)، و تسمى أيضاً Ripple Counters. هذا و يوجد نوع آخر من العدادات هي العدادات المتزامنة (Synchronous Counters)، و فيها تكون إشارة التزامن مشتركة ما بين جميع المراجيح المكونة للعداد، و دراسة هذا النوع من العدادات خارج نطاق هذا المقرر.

4-4 العد في الإتجاهين (Up/Down Counting)

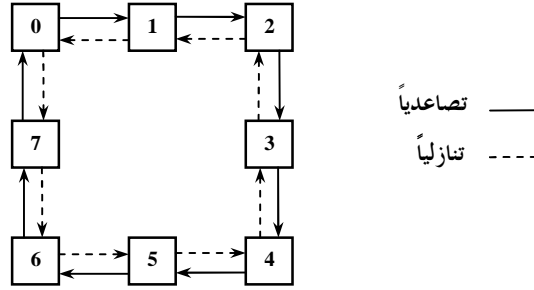
المطلوب الآن هو تصميم عدّاد يستطيع العد في الإتجاهين، أي العد تصاعدياً أو العد تنازلياً، بناء على قيمة إشارة تحكم M ، فيقوم بالعد تصاعدياً عندما تكون $M = 0$ و تنازلياً عندما تكون $M = 1$.

نعلم أن العد تصاعدياً يتطلب ربط المراجيح بطريقة معينة، و العد تنازلياً يتطلب ربطها بطريقة أخرى مختلفة. فكيف يمكن ربط المراجيح بكلا الطريقتين في وقت واحد ثم إختيار إحدهما بناء على قيمة إشارة التحكم M ؟

يتم ذلك باستخدام دوائر دامج من نوع 1 إلى 2 (2-to-1 MUX's)، مثلما فعلنا من قبل بالنسبة لمسجلات الإزاحة. حيث يدخل كل من الخرج المعكوس و الخرج غير المعكوس للمرجح لطرفي الدخل للدماج الذي يحدد أيهما يمر كإشارة تزامن للمرجح التالي بناء على قيمة الإشارة M ، فعندما تكون $M = 0$ يمر الخرج غير المعكوس و بالتالي يكون العد تصاعدياً، أما عندما تكون $M = 1$ يمر الخرج المعكوس فيكون العد تنازلياً. و الشكل التالي يوضح الدائرة المنطقية لعداد تصاعدي/تنازلي ذو ثلاثة خانات (3-bit Up/Down Counter)



و يمكن رسم مخطط الحالات (State Diagram) لهذا العدّاد بالصورة التالية

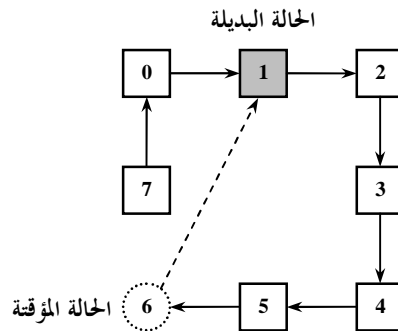


تدريب 6:

صمم عدّاداً ذو أربعة خانات (4-bit Counter) يستجيب لإشارتي تحكم M و E . الإشارة M تحدد ترتيب العد للعدّاد فيقوم بالعد تصاعدياً عندما تكون مساوية 0، و تنازلياً عندما تكون مساوية 1، و الإشارة E عبارة عن إشارة سماح (Enable) تسمح للعدّاد بالعمل عندما تكون مساوية 1 و توقف العدّاد عن العد عندما تكون مساوية 0.

5-4 العد ضمن نطاق معين

جميع العدّادات التي قمنا بتصميمها حتى الآن تمر أثناء عملية العد بجميع حالاتها، فمثلاً العدّاد ذو الثلاثة خانات يعد تصاعدياً من 0 إلى 7 أو تنازلياً من 7 إلى 0 و يمر دائماً بكل حالة من حالاته الثمانية. المطلوب الآن هو جعل العدّاد يقوم بالعد ضمن نطاق معين لا يتضمن جميع حالاته، مثلاً جعل العدّاد ذو الثلاثة خانات يقوم بالعد تصاعدياً من 1 إلى 5 فقط. يتم ذلك بالتدخل في عمل العدّاد و تغيير التسلسل الطبيعي لحالاته باستبدال حالة معينة من حالاته بحالة أخرى بصورة غير متزامنة عن طريق أطراف الدخل المباشر (Direct Inputs). فإذا بدأ العدّاد العد من الحالة 1 فإنه سيسير بالترتيب المطلوب حتى يصل إلى الحالة 5، و هي آخر حالة في تسلسل العد المطلوب، و بعد ذلك سينتقل إلى الحالة 6. يجب عندها إستبدال الحالة 6 بصورة فورية بالحالة 1 عن طريق أطراف الدخل المباشر للمراجيح. تسمى الحالة 6، و هي الحالة التي تلي مباشرة آخر حالة في نطاق العد المطلوب، بالحالة المؤقتة، في حين تسمى الحالة 1، و هي أول حالة في نطاق العد المطلوب، بالحالة البديلة. و يمكن توضيح ذلك بمخطط الحالات التالي

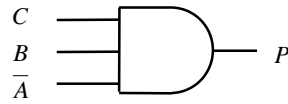


لاحظ أن الانتقال من الحالة المؤقتة إلى الحالة البديلة هو إنتقال غير متزامن لأنه يتم بصورة سريعة عبر أطراف الدخل المباشر، بحيث لا تظهر الحالة المؤقتة وإنما تظهر بدلاً عنها الحالة البديلة.

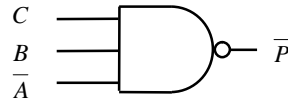
- إكتشاف وصول العدّاد للحالة المؤقتة:

يتم ذلك باستخدام بوابة AND أو بوابة NAND كما هو موضح أدناه

$$\begin{array}{c} C \quad B \quad A \\ 6 = (\quad 1 \quad 1 \quad 0)_2 \end{array} \quad \text{الحالة المؤقتة هي}$$



عند الوصول للحالة المؤقتة يصبح خرج بوابة AND P مساوياً 1.



عند الوصول للحالة المؤقتة يصبح خرج بوابة NAND \bar{P} مساوياً 0.

- إستبدال الحالة المؤقتة بالحالة البديلة:

يتم ذلك، كما ذكرنا، عن طريق أطراف الدخل المباشر (Direct Inputs) للمراجيح، حيث تستخدم الإشارة P الخارجة من بوابة AND، أو الإشارة \bar{P} الخارجة من بوابة NAND، في إجراء عمليات SET أو RESET للمراجيح بصورة مباشرة بحيث تُستبدل الحالة المؤقتة بالحالة البديلة، كما هو موضح أدناه

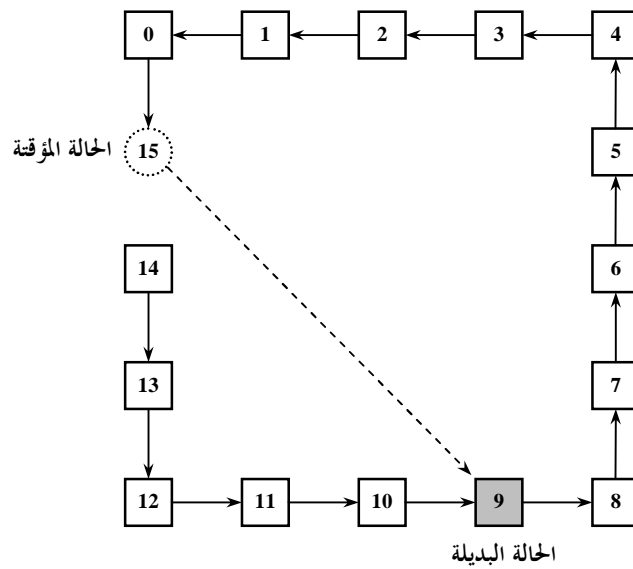
$$\begin{array}{c} C \quad B \quad A \\ 6 = (\quad 1 \quad 1 \quad 0)_2 \\ \quad \downarrow \quad \downarrow \quad \downarrow \\ 1 = (\quad 0 \quad 0 \quad 1)_2 \end{array}$$

مثال:

صمم عدّاداً يقوم بالعد تنازلياً من 9 إلى 0.

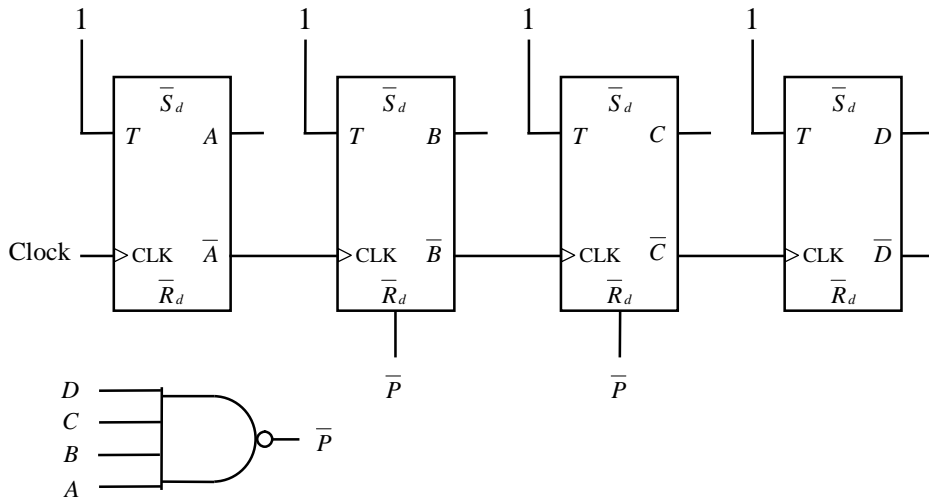
الحل:

سنقوم أولاً ببناء عدّاد ذو أربعة خانات يعد تنازلياً من 15 إلى 0، ثم نقوم بإجراء تعديل على دائرته بحيث يكون العد من 9 إلى 0. لإجراء التعديل اللازم على الدائرة يجب أن نقوم برسم مخطط الحالات للعدّاد المطلوب تصميمه لمعرفة الحالة المؤقتة و الحالة البديلة.



$$\begin{array}{ccccccc} & D & C & B & A & & \\ 15 & = & (& 1 & 1 & 1 & 1)_2 \\ & & & \downarrow & \downarrow & \downarrow & \downarrow \\ 9 & = & (& 1 & 0 & 0 & 1)_2 \end{array}$$

لاستبدال الحالة المؤقتة بالحالة البديلة نحتاج لإجراء عملية RESET للمرجحين B و C .

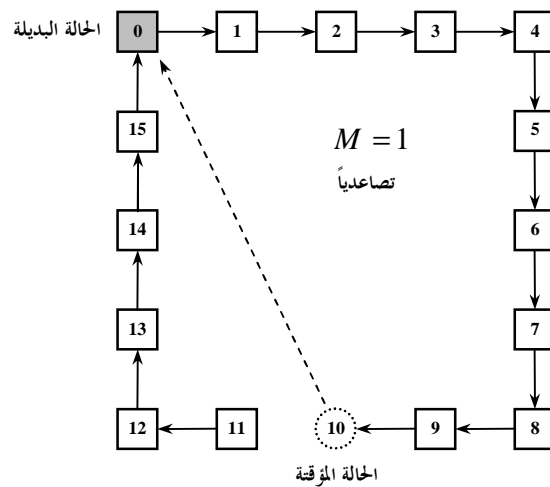


مثال:

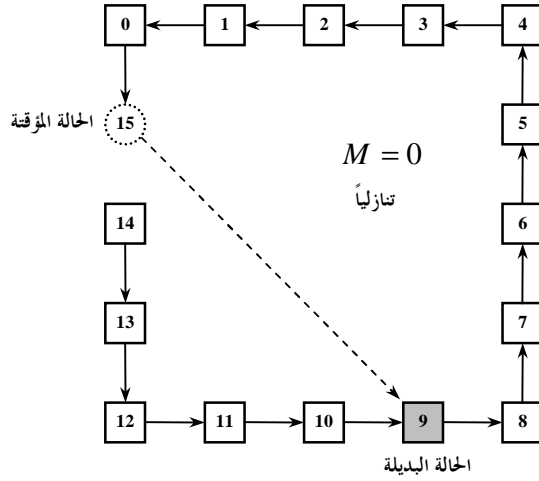
صمم عدّاداً يستجيب لإشارة تحكم M فيقوم بالعد تصاعدياً من 0 إلى 9 عندما تكون $M = 1$ و تنازلياً من 9 إلى 0 عندما تكون $M = 0$.

الحل:

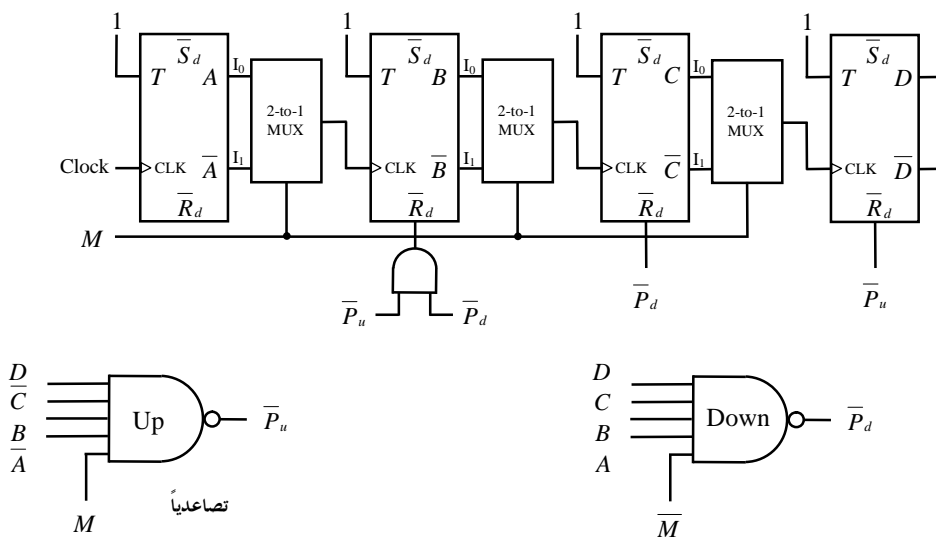
نقوم أولاً ببناء عدّاد تصاعدي/تنازلي ذو أربعة خانات، ثم نقوم بإجراء تعديل على دائرته بحيث يكون العد التصاعدي من 0 إلى 9 و العد التنازلي من 9 إلى 0. لإجراء التعديل اللازم على الدائرة يجب أن نقوم برسم مخطط الحالات للعدّاد المطلوب تصميمه لمعرفة الحالة المؤقتة و الحالة البديلة عند العد تصاعدياً و عند العد تنازلياً.



$$\begin{array}{ccccccc} & & D & C & B & A & \\ 10 & = & (& 1 & 0 & 1 & 0 &)_2 \\ & & \downarrow & \downarrow & \downarrow & \downarrow & \\ 0 & = & (& 0 & 0 & 0 & 0 &)_2 \end{array}$$



$$\begin{array}{rcccccl} & & D & C & B & A & \\ 15 & = & (& 1 & 1 & 1 & 1 &)_2 \\ & & & \downarrow & \downarrow & \downarrow & \downarrow & \\ 9 & = & (& 1 & 0 & 0 & 1 &)_2 \end{array}$$

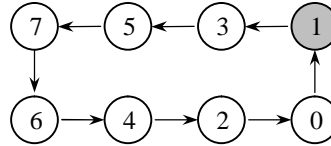


لاحظ أننا استخدمنا هنا بوابة NAND لاكتشاف الحالة المؤقتة في حالة العد التصاعدي، و خرجها هو \bar{P}_u ، و بوابة NAND أخرى لاكتشاف الحالة المؤقتة في حالة العد التنازلي، و خرجها هو \bar{P}_d . كما قمنا بإدخال إشارة التحكم M إلى بوابتي NAND (معكوسة لبوابة NAND الخاصة بالعد التنازلي و بدون عكس لبوابة NAND الخاصة بالعد التصاعدي) لضمان أن تنشط كل بوابة فقط في حالة ترتيب العد الخاص بها.

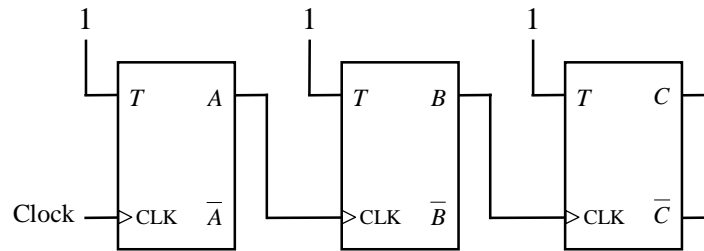
لاحظ أيضاً أننا قد استخدمنا بوابة AND في إدخال كلا الإشارتين \bar{P}_u و \bar{P}_d إلى الطرف \bar{R}_d في المرحاح B .

4-6 العد بأي ترتيب

المطلوب الآن تصميم عدّاد يقوم بالعد بتسلسل معين، و لكن بترتيب عد ليس تصاعدياً و لا تنازلياً. مثلاً مطلوب تصميم عدّاد ذو ثلاثة خانات يعد بالترتيب التالي



أبسط حل هنا هو أن نقوم بتصميم عدّاد تصاعدي ذو ثلاثة خانات (يقوم بالعد تصاعدياً من 0 إلى 7)، ثم نقوم بإدخال خرج هذا العدّاد إلى دائرة منطقية ترابطية تقوم بتحويل تسلسل العد إلى التسلسل المطلوب



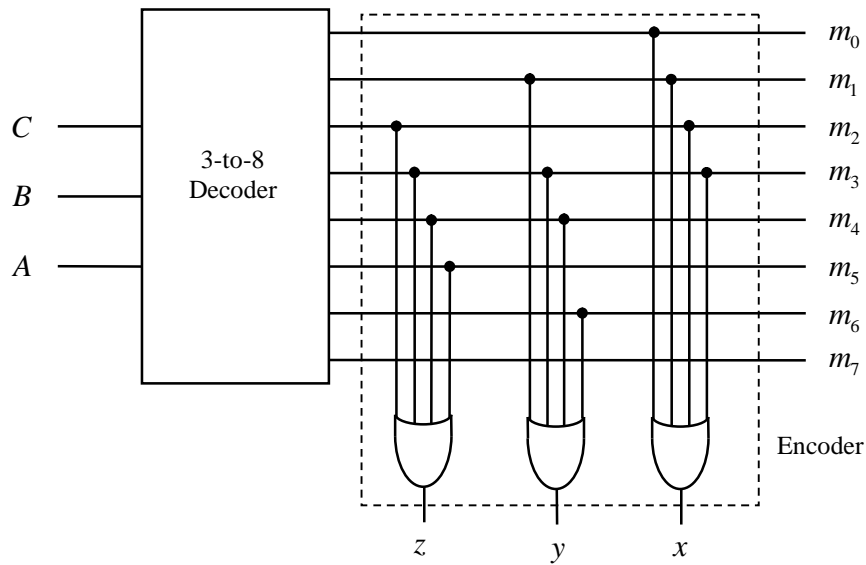
| # | C | B | A | z | y | x | Dec. |
|---|---|---|---|---|---|---|------|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 3 |
| 2 | 0 | 1 | 0 | 1 | 0 | 1 | 5 |
| 3 | 0 | 1 | 1 | 1 | 1 | 1 | 7 |
| 4 | 1 | 0 | 0 | 1 | 1 | 0 | 6 |
| 5 | 1 | 0 | 1 | 1 | 0 | 0 | 4 |
| 6 | 1 | 1 | 0 | 0 | 1 | 0 | 2 |
| 7 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

عادة ما يتم تصميم الدائرة المنطقية الترابطية هنا باستخدام فاك شفرة و مشفر (Decoder & Encoder)

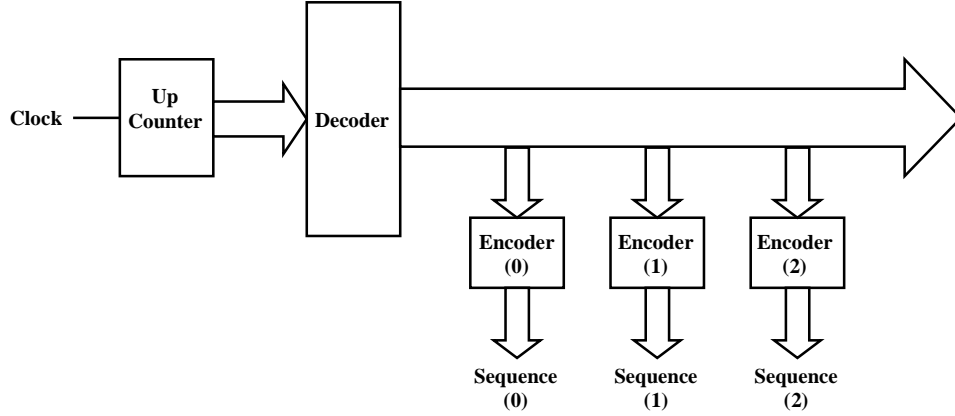
$$z = \sum m(2,3,4,5)$$

$$y = \sum m(1,3,4,6)$$

$$x = \sum m(0,1,2,3)$$



كما يمكن الحصول على مجموعة من تسلسلات العد المختلفة في وقت واحد، و ذلك باستخدام عدّاد تصاعدي واحد و فاك شفرة (Decoder) واحد و مجموعة من المشفرات (Encoders)، كل مشفر منها لتوليد تسلسل عد معين، كما هو موضح بالشكل التالي



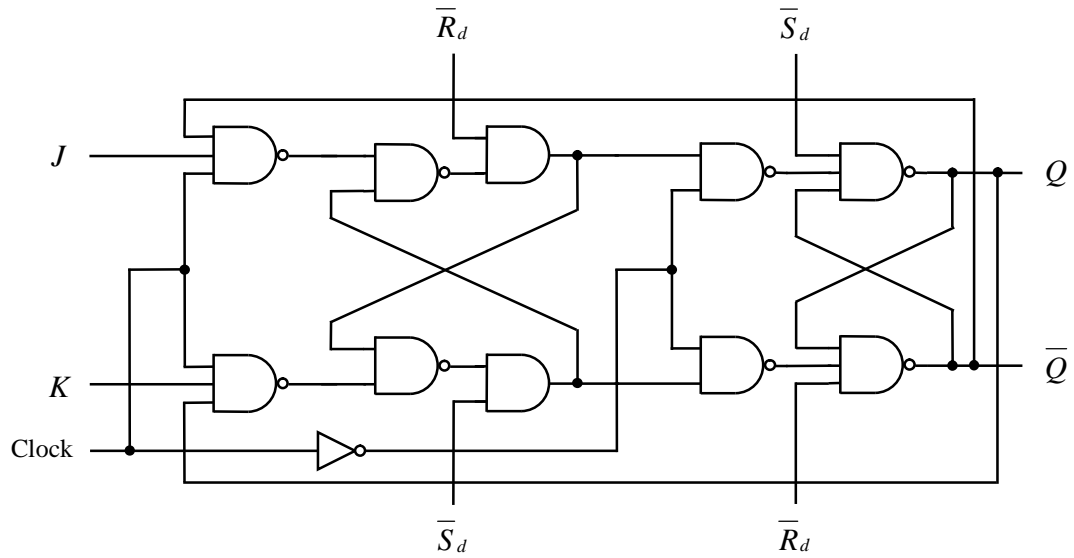
الخلاصة

قمنا في هذه الوحدة بتوضيح المقصود بالدوائر المنطقية التتابعية، و الفرق ما بينها و بين الدوائر المنطقية الترابطية. كما تعرفنا على بعض أنواع الدوائر المنطقية التتابعية الشائعة الاستخدام في الأنظمة الرقمية مثل المراجيح (Flip Flops) و المسجلات (Registers) و العدادات (Counters). حيث قمنا بعرض مختلف أنواع المراجيح، و وضعنا أهمية التزامن (Timing) بالنسبة للدوائر المنطقية التتابعية، و استخدمنا مخططات التزامن (Timing Diagrams) في تحليل تلك الدوائر. بعد ذلك إنتقلنا للمسجلات فوضحنا طريقة بنائها و العمليات المختلفة التي يمكن إجراؤها عليها من كتابة (Write) و قراءة (Read) و نقل للبيانات ما بين المسجلات (Register-to-Register Transfer). كما تعرفنا على مسجلات الإزاحة بأنواعها المختلفة و وضعنا طريقة بنائها و بعض استخداماتها. و أخيراً قمنا بعرض مبسط لبعض أنواع العدادات و وضعنا كيفية بنائها و استخدامها لتوليد تسلسل (Sequence) معين.

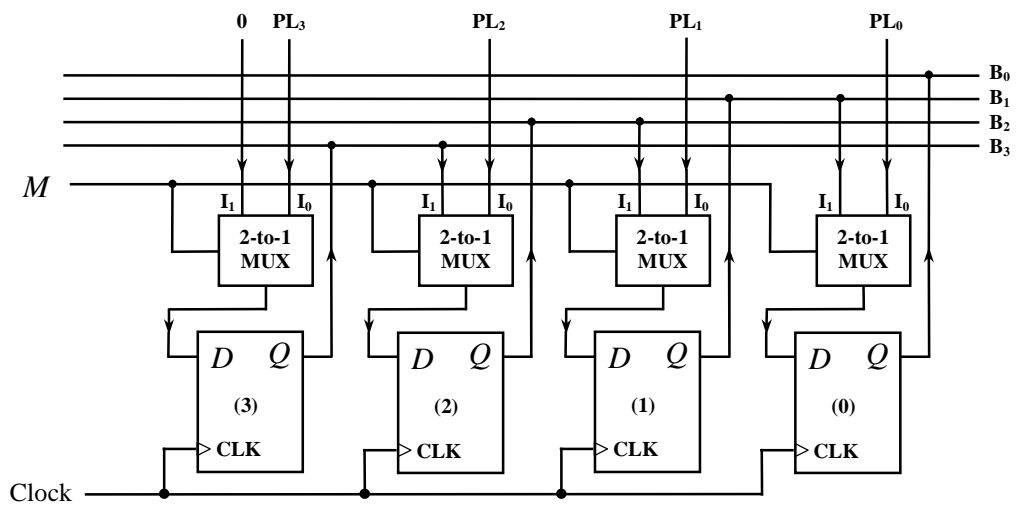
لحة مسبقة عن الوحدة التالية

نقوم في الوحدة التالية، و هي الوحدة السادسة و الأخيرة في هذا المقرر، بدراسة أهم تقنيات التخزين (Storage) المستخدمة في الأنظمة الرقمية. فنبداً بتوضيح التنظيم المنطقي للذاكرة (Memory)، ثم ننتقل لدراسة شرائح الذاكرة (Memory Chips) و أطراف التوصيل لها و طرق ربطها مع بعضها البعض. بعد ذلك ندخل إلى شريحة الذاكرة نفسها و نقوم بتوضيح البناء الداخلي لها، و نبدأ في ذلك بذاكرة الـ RAM حيث نوضح بنائها الداخلي و خصائصها و أنواعها المختلفة و استخدامات كل نوع منها، ثم ننتقل إلى ذاكرة الـ ROM و نقوم أيضاً بتوضيح البناء الداخلي لها و خصائصها و أنواعها. و في نهاية الوحدة نقوم بعرض أهم تقنيات التخزين الثانوي (Secondary Storage) مثل الأشرطة المغنطة و الأقراص المغنطة و الأقراص الضوئية.

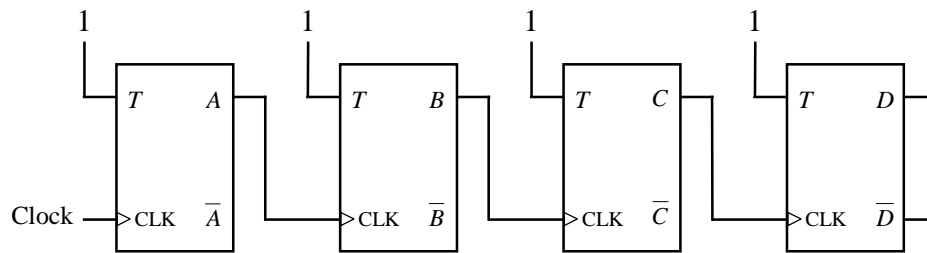
تدريب 1:



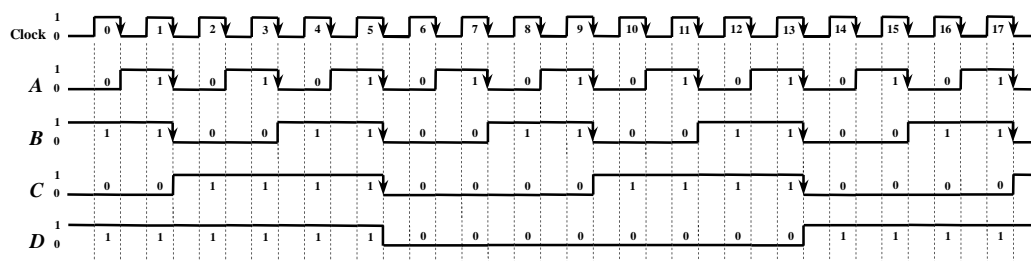
تدريب 2:



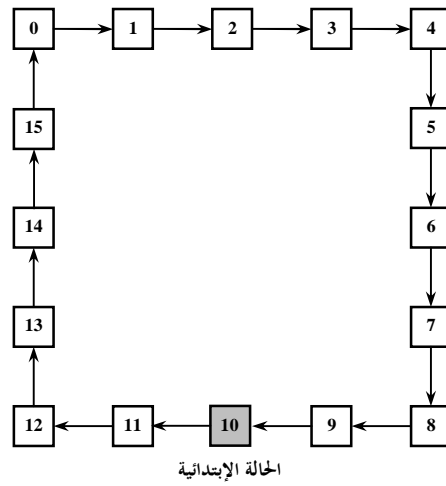
تدريپ 3:



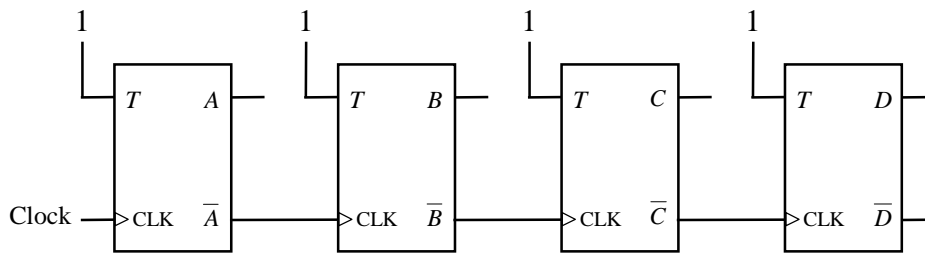
$$10 = (1010)_2$$



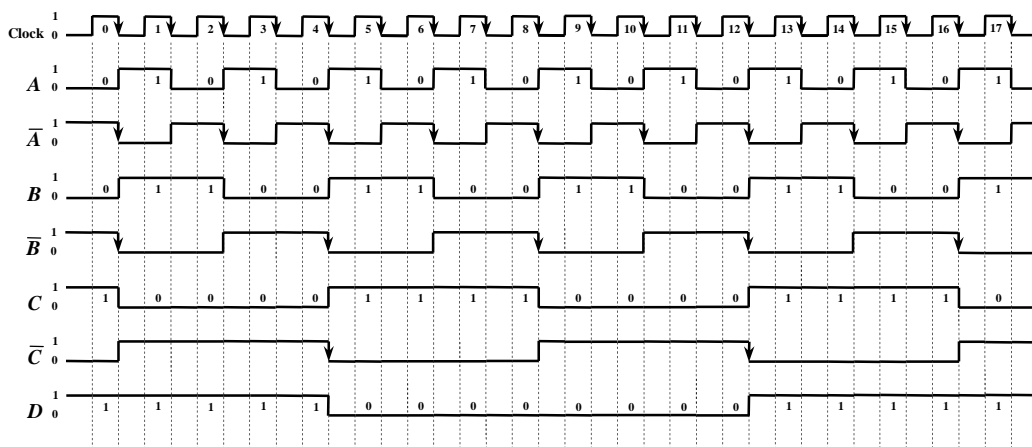
| D | C | B | A | State |
|---|---|---|---|-------|
| 1 | 0 | 1 | 0 | 10 |
| 1 | 0 | 1 | 1 | 11 |
| 1 | 1 | 0 | 0 | 12 |
| 1 | 1 | 0 | 1 | 13 |
| 1 | 1 | 1 | 0 | 14 |
| 1 | 1 | 1 | 1 | 15 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 2 |
| 0 | 0 | 1 | 1 | 3 |
| 0 | 1 | 0 | 0 | 4 |
| 0 | 1 | 0 | 1 | 5 |
| 0 | 1 | 1 | 0 | 6 |
| 0 | 1 | 1 | 1 | 7 |
| 1 | 0 | 0 | 0 | 8 |
| 1 | 0 | 0 | 1 | 9 |
| 1 | 0 | 1 | 0 | 10 |
| 1 | 0 | 1 | 1 | 11 |
| M | M | M | M | M |



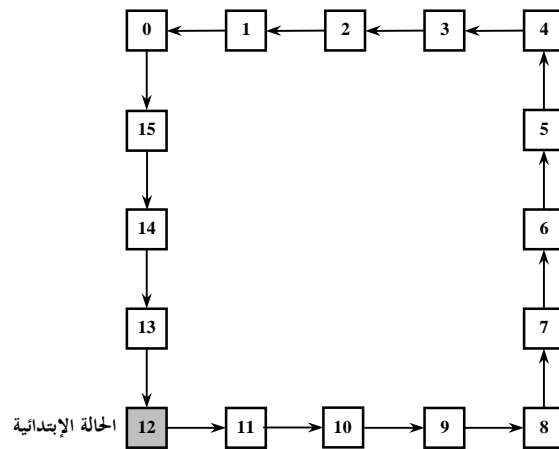
تدريب 4:



$$12 = (\begin{matrix} D & C & B & A \\ 1 & 1 & 0 & 0 \end{matrix})_2$$

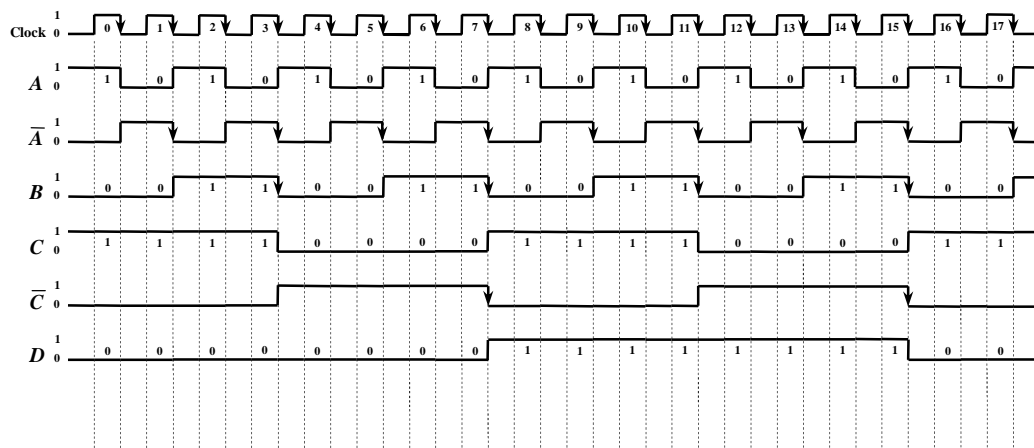


| D | C | B | A | State |
|---|---|---|---|-------|
| 1 | 1 | 0 | 0 | 12 |
| 1 | 0 | 1 | 1 | 11 |
| 1 | 0 | 1 | 0 | 10 |
| 1 | 0 | 0 | 1 | 9 |
| 1 | 0 | 0 | 0 | 8 |
| 0 | 1 | 1 | 1 | 7 |
| 0 | 1 | 1 | 0 | 6 |
| 0 | 1 | 0 | 1 | 5 |
| 0 | 1 | 0 | 0 | 4 |
| 0 | 0 | 1 | 1 | 3 |
| 0 | 0 | 1 | 0 | 2 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 15 |
| 1 | 1 | 1 | 0 | 14 |
| 1 | 1 | 0 | 1 | 13 |
| 1 | 1 | 0 | 0 | 12 |
| 1 | 0 | 1 | 1 | 11 |
| M | M | M | M | M |

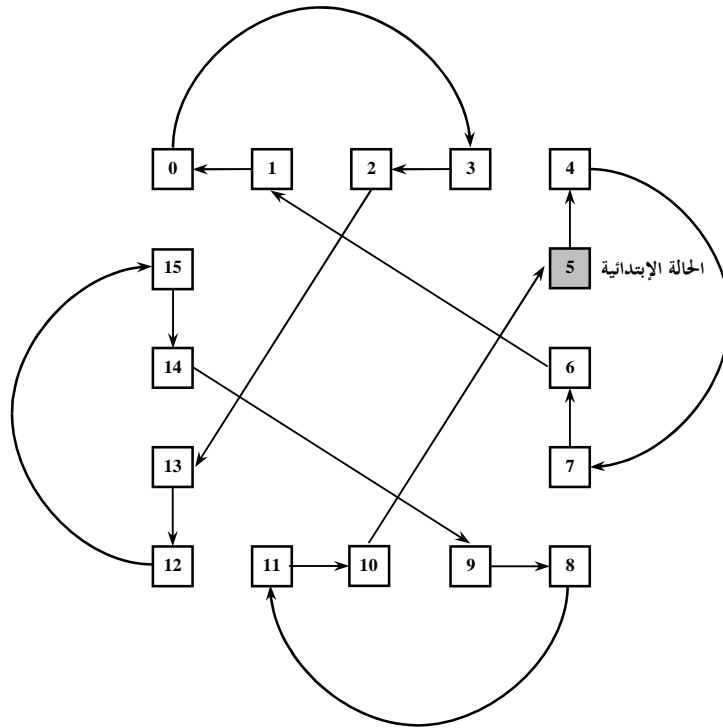


تدريب 5:

$$5 = \begin{pmatrix} D & C & B & A \\ 0 & 1 & 0 & 1 \end{pmatrix}_2$$



| D | C | B | A | State |
|----------|----------|----------|----------|----------|
| 0 | 1 | 0 | 1 | 5 |
| 0 | 1 | 0 | 0 | 4 |
| 0 | 1 | 1 | 1 | 7 |
| 0 | 1 | 1 | 0 | 6 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 3 |
| 0 | 0 | 1 | 0 | 2 |
| 1 | 1 | 0 | 1 | 13 |
| 1 | 1 | 0 | 0 | 12 |
| 1 | 1 | 1 | 1 | 15 |
| 1 | 1 | 1 | 0 | 14 |
| 1 | 0 | 0 | 1 | 9 |
| 1 | 0 | 0 | 0 | 8 |
| 1 | 0 | 1 | 1 | 11 |
| 1 | 0 | 1 | 0 | 10 |
| <hr/> | | | | |
| 0 | 1 | 0 | 1 | 5 |
| 0 | 1 | 0 | 0 | 4 |
| M | M | M | M | M |



تدريب 6:

