

تقليل استهلاك الموارد في الشبكات ضمن الرقاقة غير المتزامنة القائمة على منطق الاتفاقية الفارغة NCL

اعداد الطالبة : رهف المصطفى
اشراف : أ.د. عبد الله غندور ، د. نبيل دحدوح

الملخص

مع زيادة حجم وتعقيد الأنظمة ضمن الرقاقة (SoC)، تُعد الشبكة ضمن الرقاقة (NoC) غير المتزامنة مرشحاً مناسباً لإنشاء شبكة ربط بين مكونات النظام المتنوعة لما تتمتع به من مزايا تتراوح من زيادة كفاءة استهلاك الطاقة إلى تحسين استخدام المساحة وقابلية التوسع بالإضافة إلى التخلص من المشاكل المتعلقة بالمتزامنة وتوزيع نبضة الساعة على كامل الرقاقة. تتألف هذه الشبكات من موجّهات وخطوط نقل وواجهات ربط مع الكتل الوظيفية للنظام وتعتمد في تصميمها على تقنية الـ pipeline التي تستخدم على نطاق واسع ضمن البنى غير المتزامنة لتعزيز أداء الإنتاجية والتحكم بتدفق البيانات ضمن الشبكة في ظل غياب التزامن مع نبضة الساعة. يعتبر منطق الاتفاقية الفارغة NCL (Null Convention Logic) أحد نماذج التصميم الواعدة في مجال تنفيذ بنى منطقية غير متزامنة أكثر استقراراً وأقل استهلاكاً للطاقة، لذلك كان هناك اهتمام متزايد مؤخراً لاستخدامه في تنفيذ الشبكات ضمن الرقاقة. ولكن مشكلته الأساسية تكمن في زيادة استهلاك الموارد مقارنة بنماذج التصميم الأخرى كونه يستخدم بوابات خاصة وترميز ثنائي الأسلاك.

نقدم ضمن هذا البحث آلية لتقليل استهلاك الموارد في الشبكات ضمن الرقاقة غير المتزامنة القائمة على المنطق NCL من خلال اقتراح بنية pipeline جديدة لتنفيذ موجه الشبكة بدلاً من NCL pipeline التقليدية. تجمع هذه البنية بين مزايا تقنية RL-NCL pipeline (Register Less-) و NCL والتصميم باستخدام بوابات العتبة NCL القياسية بحيث نستغني عن الحاجة لبوابات MTNCL (Multi-threshold NCL) المستخدمة في RL-NCL pipeline وبالتالي تصبح

تقليل استهلاك الموارد في الشبكات ضمن الرقابة غير المتزامنة القائمة على منطق الاتفاقية الفارغة NCL

قابلة للتحقيق على الـ (Field Programmable Gate Arrays) FPGAs أو باستخدام بوابات المنطق NCL القياسية.

تمت المحاكاة باستخدام أداة المحاكاة Cadence® إلى جانب أداة التصميم بمساعدة الحاسب الأكاديمية VTR (Verilog-to-Routing) حيث بينت نتائج المقارنة انخفاضاً ملحوظاً في استهلاك الموارد عند تنفيذ موجه الشبكات ضمن الرقابة باستخدام البنية المقترحة بنسبة 27.32 % مقارنة بالبنية المُنفذة باستخدام NCL pipeline التقليدية وهذا الانخفاض أدى بدوره إلى تحسن في زمن تأخير الموجه وتوفير استهلاك الطاقة.

الكلمات المفتاحية: دارات غير متزامنة، الشبكة ضمن الرقابة، منطق الاتفاقية الفارغة، pipeline، FPGA، RL-NCL.

Reducing Resource Consumption in Asynchronous Network-on-Chip Based on Null Convention Logic (NCL)

Abstract

As system on a chip (SoC) complexity grows, asynchronous networks on chip (NoC) emerge as effective solutions for interconnecting components, offering benefits like power efficiency, scalability, and elimination of synchronization issues. These networks consist of routers, links, and interfaces with system's functional blocks, and using pipeline technique, which is widely used in asynchronous architectures to enhance throughput performance and control data flow within the network in the absence of clock synchronization.

Recently, there has been increasing interest in using Null Convention Logic (NCL) for implementing Networks-on-Chip, as it is considered one of the promising design paradigms for constructing low-power robust asynchronous circuits. However, it suffers from increased resource consumption compared to other design paradigms, due to its use of specialized gates and dual-rail encoding.

This research introduces a resource-efficient method for implementing NCL-based NOC router architectures using Register-Less-NCL pipeline technique, and propose new method to design NCL gates, which eliminating the need for Multi-threshold NCL (MTNCL) gates in this pipeline, and can be implemented even in Field Programmable Gate

Arrays (FPGAs) or using the standard NCL cells method. Simulation results show that The new design of NOC router using the proposed RL–NCL pipeline architecture was able to achieve a significant reduction in resource consumption by 27,32% compared with the conventional NCL pipeline architecture. This reduction in resource consumption, in turn, leads to an improvement in router latency and power savings.

Keywords: Asynchronous Circuits, NOC, NCL, FPGA, Pipeline, RL–NCL.

1. مقدمة:

يزداد تعقيد الأنظمة الرقمية مع استمرار زيادة عدد الترانزستورات الموضوعة على رقاقة واحدة وفق قانون مور، حيث أصبح في الوقت الحاضر عدد الترانزستورات ضمن الرقاقة الواحدة في نطاق المليارات، كما يتم دمج المزيد من المكونات ضمن الرقاقة، مثل نوى المعالجة ومصفوفات الذاكر لتكوين الأنظمة ضمن الرقاقة (SOC (System on Chip)، وبالتالي أصبح التواصل بين هذه المكونات عاملاً محدداً لتحقيق أداء عالٍ واستهلاك منخفض للطاقة.

في بداية القرن العشرين ظهر مفهوم الشبكات ضمن الرقاقة (NOC (Network on Chip) لحل مشكلة الاتصالات والتعقيد المتزايد ضمن الرقاقة، وهي مشابهة للشبكات المستخدمة في المجالات الأخرى حيث تتكون من عقد توجيه وقنوات تصل بينها، ولكن مع مرور الوقت أصبحت خاصية التزامن في الشبكات ضمن الرقاقة عاملاً مقيداً في بعض الأنظمة حيث تزداد صعوبة توزيع نبضات الساعة كلما أصبحت الأنظمة ضمن الرقاقة SOC أسرع ومعقدة أكثر ومساحتها أكبر [1] وهذا التحدي تواجهه أيضاً لوحات مصفوفات البوابات المنطقية القابلة للبرمجة FPGA (field programmable gate arrays) مما يجعل التصميم غير المتزامنة (أو على الأقل التواصل بين الكتل الوظيفية ضمن الرقاقة بشكل غير متزامن) تأخذ حيزاً كبيراً من الاهتمام لأنها تخفف العبء الكبير الناتج عن توزيع نبضات الساعة الرئيسية. كما أنها توفر مزايا أخرى مثل انخفاض التداخل الكهرومغناطيسي (electromagnetic interference) EMI وانخفاض استهلاك الطاقة وإمكانية إعادة استخدام التصميم بالاعتماد على التشغيل الذاتي [2].

هناك العديد من النماذج المستخدمة لتصميم البنى غير المتزامنة وهي تنقسم بشكل عام إلى مجموعتين نماذج التأخير المحدود (Bounded Delay) BD ونماذج غير حساسة للتأخير DI (Delay-Insensitive). في نماذج التأخير المحدود يتم تقييم التأخيرات في كل من البوابات والأسلاك وتحديدها وفقاً لأسوأ السيناريوهات لتجنب الأخطاء. من ناحية أخرى، لا تضع النماذج غير الحساسة للتأخير أي افتراضات حول التأخير في منطقتها أو الترابط فيما بينها وتعتمد على بنيتها الداخلية وآلية ترميز البيانات للتحكم بتدفق البيانات دون حدوث أخطاء. في البداية كان الاهتمام موجه نحو نماذج التأخير المحدود كونها تستخدم بنية مشابهة إلى حد ما لبنية التصميم المتزامنة ولكن مع تزايد مشاكل التوقيت والمسارات الحرجة أصبحت التصميم غير الحساسة

للتأخير ملاذاً آمناً لبناء أنظمة ضمن الرقابة أكثر استقراراً وتوفيراً للطاقة وخاصة في مجال الشبكات ضمن الرقابة على الرغم من استهلاكها الكبير للموارد والمساحة مقارنة بالتصاميم الأخرى.

يعد منطق الاتفاقية الفارغة NCL [3] من أكثر النماذج الواعدة في مجال التصاميم غير الحساسة للتأخير كونه متكامل منطقياً (فهو ليس بحاجة لعناصر غير منطقية مثل الساعة أو عناصر تحكم أو خطوط تأخير لضبط وإكمال العملية)، و مستقل محلياً، متزامن ذاتياً، وقادر على اكتشاف الأخطاء تلقائياً بالاعتماد على بنيته التركيبية.

يهدف ضمن هذا البحث إلى تحسين بنى الشبكات ضمن الرقابة القائمة على المنطق NCL وتقليل المشاكل المتعلقة بهذا المنطق من خلال تعديل تقنية الـ pipeline المستخدمة ضمن هذه الشبكات لتصبح أقل استهلاكاً للموارد، وبناءً على ذلك نستعرض بداية بنية الشبكات ضمن الرقابة غير المتزامنة ومفهوم تقنية الـ pipeline ودورها ضمن هذه الشبكات و نستذكر خصائص المنطق NCL، وفي القسم الثاني نستعرض أبرز الدراسات المرجعية في مجال الـ pipeline، وأخيراً نقترح آلية لتحسين بنية الـ pipeline القائمة على المنطق NCL بهدف استخدامها في تحقيق شبكات ضمن الرقابة أقل استهلاكاً للموارد وتم عرض نتائج تنفيذ موجه الشبكة باستخدام التقنية المقترحة ومقارنتها ببنية الموجه المُنفذ باستخدام NCL pipeline التقليدي في القسم الأخير من هذه الدراسة.

2. هدف البحث:

أغلب الأنظمة ضمن الرقابة حالياً تستخدم الشبكات كوسيلة للتواصل بين مكونات النظام نظراً لأدائها الفعال من حيث الإنتاجية والموثوقية والأمان مقارنة بالوصلات التقليدية ولكنها تشكل عبء على النظام من ناحية استهلاك الموارد، يهدف هذا البحث إلى تقليل استهلاك الموارد من قبل الشبكات ضمن الرقابة غير المتزامنة القائمة على المنطق NCL مع المحافظة على أداءها من خلال تحقيق بنية الموجه باستخدام Register-Less-NCL pipeline ولكن تصميم هذا الـ pipeline مخصص بالكامل حيث تم تنفيذه على مستوى الترانزستور بالاستفادة من بوابات المنطق MTNCL وبالتالي غير قابل للتنفيذ باستخدام بنى المنطق NCL القابلة لإعادة البرمجة

لذلك تم اقتراح بنية جديدة تجمع بين مزايا تقنية RL-NCL pipeline والتصميم باستخدام بوابات NCL القياسية.

3. الشبكات ضمن الرقابة غير المتزامنة:

تتكون الشبكة ضمن الرقابة من عقد التوجيه وقنوات تصل فيما بينها. تحدد هيكلية الشبكة ترتيب هذه العقد وكيفية ربطها ببعضها بواسطة القنوات، ويكون اختيار هيكلية الشبكة عادةً هو الخطوة الأولى في تصميم الشبكة حيث تعتمد استراتيجيات التوجيه وآلية التحكم بالتدفق بشكل كبير عليها، وهناك هيكليات عديدة للشبكات ضمن الرقابة تختلف باختلاف الغرض من الرقابة بهدف الحصول على أفضل أداء للشبكة.

تنقسم الشبكات ضمن الرقابة حسب تصميمها إلى شبكات متزامنة بالكامل وشبكات متعددة التزامن وشبكات غير متزامنة. تتضمن الشبكة المتزامنة بالكامل ساعة واحدة موزعة على كامل شبكة الاتصالات والتي قد تعمل بمعدل ساعة مختلف عن الوحدات الوظيفية التي تربط بينها ونظراً لأن الشبكة تمتد عبر الرقابة فأن توزيع نبضات الساعة يمثل تحدياً كبيراً، فوفقاً للأبحاث تستهلك هذه الشبكات ما يصل إلى 30% من ميزانية طاقة الرقابة بسبب توزيع الساعة [4]، كما أنها ذات معدل إنتاجية محدود وعادةً ما تكون أبطأ ب 2-3 مرات من الوحدات الوظيفية مما يؤثر بشكل سلبي على أداء عملية نقل البيانات، أما في الشبكات متعددة التزامن يكون لكل موجه تردد تشغيل خاص يمكن أن يساوي تردد تشغيل الطرفية المتصل بها، مما يقلل من استهلاك الطاقة الديناميكي عند مقارنته بالشبكات أحادية التردد، ولكن يجب إضافة واجهات تزامن بين الموجهات وبين كل موجه والوحدة الوظيفية التي تعمل ضمن مجال تزامن مختلف.

بالنسبة للشبكات ضمن الرقابة غير المتزامنة تستخدم إشارات المصافحة لضبط تدفق البيانات ضمن الشبكة بدلاً من تزامن نبضة الساعة مما يسمح بتكامل أكثر مرونة لمكونات النظام ذات خصائص التزامن المختلفة، حيث يمكن أن تستخدم كشبكة اتصال لوحدات وظيفية غير متزامنة أو متزامنة كما في أنظمة GALS (Globally Asynchronous Locally Synchronous)) مما يلغي الحاجة إلى إدارة الساعة عبر شبكة واسعة، ويقتصر التزامن على عقدة المصدر وعقدة الوجهة فقط وبالتالي تقلل التأخيرات الزمنية الناتجة عن واجهات التزامن، إلى جانب تقديم استهلاك

منخفض للطاقة عند مقارنتها بالشبكات المتزامنة، حيث لا يتم استهلاك أي طاقة ديناميكية تقريباً عندما تكون الشبكة في وضع الخمول.

يختلف تصميم الشبكات غير المتزامنة باختلاف نموذج التصميم المستخدم الذي يحدد من خلاله آلية عمل بروتوكول المصافحة وطريقة ترميز البيانات، ولكن جميع هذه التصميمات تعتمد آلية العمل نفسها تقريباً وهي أيضاً المتبعة في تصميم معظم الدارات الرقمية والتي تقوم على تقسيم الدارة إلى مراحل مع إدخال مسجلات بينها للاحتفاظ بالقيم الناتجة عن كل مرحلة وتمهيداً للمرحلة التالية بهدف زيادة الإنتاجية وهذا ما يعرف باسم تقنية الـ pipeline .

4. تقنية الـ pipeline:

تستخدم الـ pipeline على نطاق واسع لتعزيز أداء الإنتاجية في الأنظمة الرقمية عالية السرعة مثل المعالجات الحديثة المستخدمة للأغراض العامة، ووحدات الرسوم، والدارات الرقمية الخاصة بتطبيقات معينة، مثل الوسائط المتعددة، وكذلك هو الحال بالنسبة للموجهات في معظم أنظمة الشبكات ضمن الرقابة عالية الأداء. يقسم تنفيذ الـ pipeline النموذجي الوحدات الوظيفية المعقدة إلى وحدات أصغر، حيث تحقق رزم البيانات ضمن الـ pipeline على التوالي، ويتم معالجتها من قبل الوحدات الجزئية المختلفة بالوقت نفسه.

على الرغم من أن فكرة المعالجة المتوازية لرزم الدخل المتعددة هي نفسها، إلا أن الـ pipeline غير المتزامنة تتمتع بعدة مزايا أساسية مقارنة بالـ pipeline غير المتزامنة [5] أهمها:

(1) في الـ pipeline المتزامنة، يتم حساب تردد الساعة من خلال المسار الحرج للدائرة وتعمل جميع المراحل بنفس المعدل، بينما في التصميم غير المتزامن، تعمل كل مرحلة وفقاً لمسارها الحرج الخاص بها، مما يقلل من زمن التأخير.

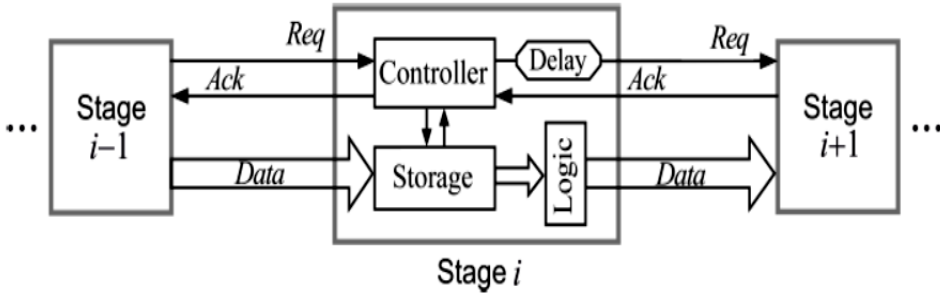
(2) يمكن أن تتلقى مراحل الـ pipeline غير المتزامنة عدد متغير من رزم البيانات في أي وقت، ولهذا السبب تتم معالجة كل من هذه الرزم عند توفرها، بينما في المتزامنة لضمان سير العملية بشكل صحيح يجب أن تصل البيانات الجديدة خلال فترات زمنية محددة مسبقاً.

(3) التحكم المتأصل في التدفق بسبب بروتوكولات المصافحة التي تستخدمها الأنظمة غير المتزامنة.

(4) يكون استهلاك الطاقة الديناميكية عند الطلب، أي فقط عندما يكون هناك بيانات يجب معالجتها وغير ذلك تكون الدارة خاملة.

وقد أدى هذا إلى العديد من المنتجات الناجحة مثل IBM-TrueNorth و IBM-FIR Filter و Intel/Fulcrum و Intel RAPPID و Achronix-Fastest FPGAs و ورقائق مبدلات Ethernet [6-9].

يبين الشكل (1) بنية توضيحية لتقنية الـ pipeline غير المتزامنة بشكلها الخطي، حيث ترتبط المراحل المتتالية بقناة اتصال تمتد عبر إشارات المصافحة، والتي تتكون عادةً من الطلب (Req) وتأكيد الطلب (Ack)، بالإضافة إلى البيانات (Data) التي يتم إرسالها.



الشكل (1): بنية توضيحية لتقنية الـ pipeline غير المتزامنة.

بشكل عام، تستقبل المرحلة i البيانات من المرحلة السابقة (المرحلة $i-1$)، ويتم تخزينها ومعالجتها قبل إرسالها إلى المرحلة اللاحقة (المرحلة $i+1$). وحدة التحكم داخل "صندوق المرحلة" هي التي تلعب دور تنسيق إشارات المصافحة بالإضافة إلى تنظيم التوقيينات المحلية لتخزين البيانات و/أو معالجتها.

افتراضنا في البنية الموضحة في الشكل (1) استخدام نموذج نقل البيانات المجمع ذات التأخير المحدود، حيث تم إضافة عنصر تأخير صريح إلى مسار الطلب لمطابقة أسوأ تأخير ضمن كتلة المنطق (Logic). إذا لم تكن هناك حاجة لجزء المنطق، فإن البنية تصبح عبارة عن نظام FIFO (First-In-First-Out) غير متزامن.

تجدر الإشارة إلى أن الدارة العملية تعتمد عادةً على كل من الهياكل الخطية وغير الخطية لتقنية الـ pipeline، مثل التفرعات والوصلات والمحكمات.

1.4. نظرة عامة على تنفيذ دارة ال pipeline غير المتزامنة:

يتطلب التنفيذ الخاص لبنية ال pipeline الموضح في الشكل (1) حل مجموعة من مشكلات التصميم، بدءًا من الاعتبارات عالية المستوى، مثل مخططات ترميز البيانات وبروتوكولات إشارات المصافحة، ووصولًا إلى خيارات مستوى الدارة، مثل عائلات المنطق وطرق تخزين البيانات، وذلك لتحقيق الأهداف المتعلقة بمساحة الدارة والسرعة واستهلاك الطاقة. في هذا القسم من البحث، نناقش هذه القضايا التصميمية من خلال دراسة قنوات الاتصال أولًا، ثم دراسة بنية مرحلة ال pipeline.

1.1.4. قنوات الاتصال:

أ. ترميز البيانات:

يعتمد تصميم ال pipeline غير المتزامن بشكل رئيسي على الطريقة المستخدمة في ترميز البيانات، حيث هناك طريقتان رئيسيتان لترميز البيانات في الدارات غير المتزامنة هما (أ) الترميز ثنائي الأسلاك وهو المستخدم غالبًا في نماذج التصميم غير الحساسة للتأخير [10-14] و(ب) ترميز البيانات المجمع أحادية السلك المستخدم عادةً في نماذج التأخير المحدود [15-20]. في الترميز ثنائي الأسلاك، يتم استخدام سلكين لترميز كل بت (أحدهما للبيانات والآخر للإشارة إلى صحتها). من ناحية أخرى، يحافظ ترميز البيانات المجمع أحادية السلك، كما هو موضح في الشكل (1)، على تطبيقات المنطق المتزامن أحادي السلك التقليدية المستخدمة ضمن مسار البيانات. ومع ذلك، فإنه يُدخل إشارات مصافحة إضافية للإشارة إلى صحة البيانات، كما يستخدم أيضاً تقنية تُعرف باسم مطابقة التأخير [21]، أي إدراج عنصر تأخير صريح ضمن مسار إشارة الطلب Req يتم تنفيذه عادةً بواسطة سلسلة عاكس أو نسخة طبق الأصل عن المسار الحرج ضمن كتلة المنطق لمطابقة أسوأ تأخير لهذه الكتلة ضمن المرحلة الواحدة.

بشكل عام، تتطلب الدارات ذات الترميز ثنائي الأسلاك مساحة إضافية. ومع ذلك، فهي تُمكن من قياس التأخير في الوقت الفعلي من خلال استشعار حالة الأسلاك باستخدام دارات كشف الإكمال [21]، وبالتالي تتمتع بمثانة توقيت أكبر، بينما تُعدّ دارات البيانات المجمع أسهل للمصممين ذوي التفكير المتزامن، على الرغم من أن هذه الدارات يجب أن تضمن هوامش كافية لتأخير مسار البيانات في أسوأ الحالات في ظل تغيرات PVT (process, voltage and temperature).

ب. بروتوكولات المصافحة:

يشير بروتوكول المصافحة إلى تسلسل انتقالات إشارات المصافحة لتسهيل نقل البيانات، حيث يشير الانتقال إما إلى الحافة الصاعدة (+) أو الحافة الهابطة (-) لإشارة Req أو Ack. بروتوكولات إشارات المصافحة الرئيسية الثلاثة هي ثنائي الطور، ورباعي الطور، وأحادي المسار. بروتوكول المصافحة ثنائي الطور [13,15,20,22]، ويُعرف أيضاً باسم بروتوكول عدم العودة إلى الصفر (NRZ) لأن كل انتقال ل Req و Ack ينقل رسالة نشطة، على سبيل المثال، يُشير كل من Req+ و Req- إلى أن البيانات المُجمّعة أصبحت صالحة، بينما يُشير كل من Ack+ و Ack- إلى أن البيانات قد استُهلكت.

أما بالنسبة لبروتوكول المصافحة رباعي الطور [16,20,22]، أو بروتوكول العودة إلى الصفر (RZ) فهو يستخدم حافة واحدة فقط لكل إشارة مصافحة لنقل الرسائل النشطة، بينما يُعاد ضبط الحافة الأخرى ببساطة. تُمكن التركيبات المختلفة للحواف النشطة ل Req و Ack من ظهور أنواع فرعية مختلفة من البروتوكولات رباعية الطور.

يدمج البروتوكول أحادي المسار [23] بين طلب الإرسال req والتأكيد ack ضمن سلك واحد حيث يُصدر المرسل طلباً عن طريق إنشاء انتقال على السلك، ويُقرّ المستقبل بالاستلام عن طريق عكس هذا الانتقال. لذلك، يمتلك هذا البروتوكول ميزات كل من البرتوكولين السابقين (ثنائي ورباعي الطور).

فيما يتعلق بكفاءة الاتصال لبروتوكولات المصافحة المذكورة أعلاه، يُعتقد عموماً أن البروتوكولات ثنائية الطور وأحادية المسار تُمكن من معدل بيانات أعلى، أو زمن دورة أقصر لدارات ال pipeline، مقارنةً بالبروتوكولات رباعية الطور [5]، ولكن قد تتطلب الدارات القائمة على بروتوكولات المصافحة ثنائية الطور دارات تحويل إضافية تتفاعل مع التخزين والمنطق القائم على المستوى، وقد تُسبب صعوبات في تصحيح الأخطاء [24].

2.1.4. مرحلة ال pipeline (تنفيذ تخزين البيانات والمنطق ووحدة التحكم):

بعد تحديد مخطط ترميز البيانات وبروتوكول المصافحة، يحين الوقت المناسب للنظر في تنفيذ الدارة ضمن مرحلة الـ pipeline، حيث يتم بشكل عام اتباع تسلسل التصميم التالي المكون من ثلاث خطوات:

1. تحديد عناصر تخزين البيانات:

يعود اختيار عناصر التخزين إلى نموذج التصميم المستخدم حيث يمكن أن تكون عناصر تخزين البيانات المستخدمة ضمن الـ pipeline غير المتزامنة هي نفسها المعتمدة في الدارات التسلسلية المتزامنة، مثل المواسك الحساسة للمستوى والقلابات الحساسة للحافة، أو يمكن اختيار عناصر تخزين بيانات مصممة خصيصاً لبنى pipeline غير المتزامنة، مثل مواسك النقاط التمير المستخدمة في الـ micro-pipeline.

2. اختيار نمط المنطق:

يمكن استخدام كل من نمطي المنطق الساكن والديناميكي لتنفيذ مسار البيانات ضمن بنى الـ pipeline غير المتزامنة التي تعتمد في تصميمها على نموذج التأخير المحدود. يتميز المنطق الساكن بأنه خالٍ من تداخل إشارات المصافحة، كما هو موضح في الشكل (1). أما المنطق الديناميكي، فيتطلب تنسيق مرحلتي الشحن المسبق والمعالجة بواسطة وحدات تحكم محلية، ولكنه قد يُحسن أداء الدارة بشكل أكبر على حساب مخطط تحكم أكثر تعقيداً. بالمقارنة مع نموذج التأخير المحدود، فإن تصميم مسار بيانات باستخدام النموذج غير الحساس للتأخير DI يتطلب جهداً أكبر، إذ يجب تنفيذ شكل مناسب من المؤشرات للتحكم في انتقال البيانات بين المراحل، إما عبر المنطق نفسه أو من خلال دارات كشف إكمال إضافية حيث توجد أشكال مختلفة للتصميم غير الحساس للتأخير مثل منطق الاتفاقية الفارغة (NCL) القائم على بوابات العتبة وتوليف الحد الأدنى غير الحساس للتأخير (DIMS). علاوة على ذلك، تتوفر تقنيات تحسين منطقي لتقليل تعقيد الدارة مع الحفاظ على الاستقرار.

3. تصميم وحدة التحكم:

تلعب وحدة التحكم بالمصافحة دوراً رئيسياً في تنسيق إشارات المصافحة بين المراحل المتجاورة لـ pipeline وتعتبر من أكثر مجالات البحث اهتماماً في تصميم الدارات غير المتزامنة، حيث يجب أن يتبع سلوك وحدة التحكم المنفذة بدقة علاقات التوقيت المنسقة جيداً مع البيئة،

وألا تُؤلّد أي أعطال زمنية (ما يُسمى بالخالي من المخاطر) بخلاف كتلة المنطق ضمن مسار البيانات، والتي يمكن أن تتسامح بحدوث بعض الأعطال المؤقتة، كما يعتمد التشغيل الصحيح لدارة التحكم غير المتزامنة على افتراضات نموذج التأخير الخاص بها.

5. منطق الاتفاقية الفارغة NCL :

كما ذكرنا في مقدمة هذا البحث يمكن تصنيف الدارات غير المتزامنة وفقاً لنموذج تأخيرها إلى دارات ذات تأخير محدود (BD) ودارات غير حساسة للتأخير (DI). تعتمد الأولى في تصميمها على قيم تأخير محددة وفترات زمنية محدودة للعمل بشكل صحيح. إذا لم تلتزم الدارة بهذه الفترات، فلا يمكن ضمان التشغيل الصحيح. لذلك تُدرج ضمن الدارة تأخيرات مطابقة أكبر من تأخيرات المسارات الحرجة.

أما الدارات غير الحساسة للتأخير (DI) فهي تعمل بشكل صحيح بغض النظر عن تأخير البوابات والأسلاك. وبالتالي، يُعد نموذج DI الأكثر استقراراً من بين النماذج غير المتزامنة ولكن صعب التحقيق من الناحية العملية لذلك يعتبر النموذج شبه غير الحساس للتأخير (QDI) هو النموذج الأكثر استقرار الذي يُمكن تطبيقه في معظم الدارات. حيث تعمل دارات QDI بشكل صحيح بغض النظر أيضاً عن تأخير البوابات والأسلاك، ويقتصر القيد الزمني الوحيد على بعض تأخيرات الأسلاك، مثل افتراضات التفرعات المتزامنة والتغذية العكسية.

ويهدف تحقيق استقرار أكبر يتم استخدام رموز تدعم النموذج شبه غير الحساس للتأخير، مثل ترميز الأسلاك المزدوجة وهو الترميز الأكثر شيوعاً في تصميم دارات NCL، حيث يُمثل كل بت من البيانات بسلكين، سلك يعبر عن الصفر منطقي (أو False) وسلك يعبر عن الواحد منطقي (أو True) وفقاً للجدول (1) المبين أدناه حيث لا يمكن أن يكون كلا السلكين واحد منطقي في نفس الوقت، لذا يتم فصل الانتقال من قيمة بيانات صالحة إلى أخرى بقيمة NULL (قيمة فارغة).

الجدول (1): التعبير وفق الترميز مزدوج الأسلاك.

A ¹	A ⁰	A
0 v	0 v	NULL
0 v	1 v	Logic 0 (data)
1 v	0 v	Logic 1 (data)
1 v	1 v	Not allowed

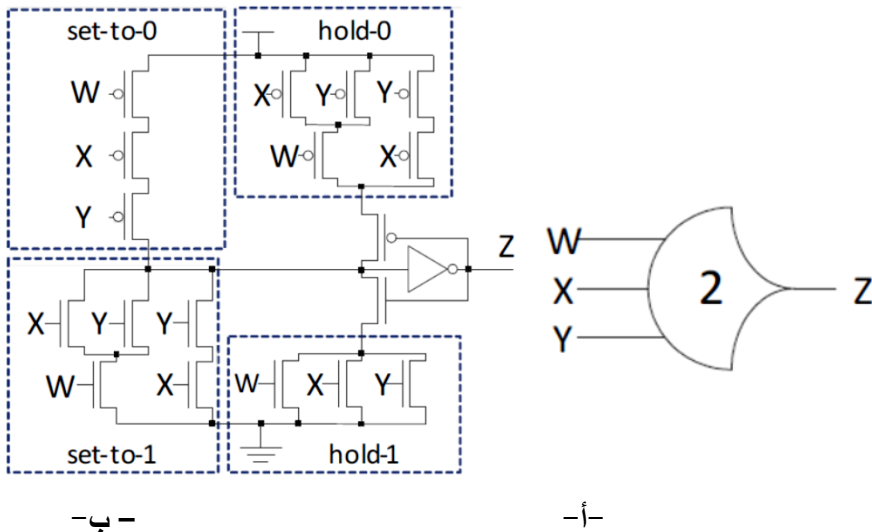
يمكن أن تستخدم الدارات غير المتزامنة كما ذكرنا سابقاً ضمن هذا البحث بروتوكول ثنائي الطور أو رباعي الطور للتعبير عن حالة معينة على أسلاكها، ونظراً لطبيعة الدارات ثنائية الأسلاك فهي تستخدم عادةً بروتوكول رباعي الطور. إلا أن الوضع مختلف قليلاً بالنسبة لدارات NCL حيث يُشير الانتقال من منخفض إلى مرتفع إلى إقرار باستلام بيانات صالحة، ويُشير الانتقال من مرتفع إلى منخفض إلى إقرار بقيمة NULL.

منطق الاتفاقية الفارغة NCL [3] هو نموذج منطقي غير متزامن شبه حساس للتأخير (QDI) حيث يكون التحكم متأصلاً في طريقة التعبير عن البيانات، وبالتالي فهو يوفر تصميمات جاهزة للوصل والتشغيل، ولا يتطلب تحليلاً لأسوأ حالات التأخير. تم استخدام NCL بنجاح في عدد من المنتجات التجارية، بما في ذلك وحدات التحكم الدقيقة والمنتجات الطبية المضمنة ومحركات التشفير لتطبيقات البطاقات الذكية [26]. بالإضافة إلى ذلك، تم تطوير العديد من أدوات أتمتة التصميم الإلكتروني (EDA) الخاصة بالمنطق NCL [27]، وفي السنوات الأخيرة، تم استخدام NCL في مجموعة متنوعة من التطبيقات، بما في ذلك تصميم الدارات منخفضة الطاقة [28]، ودارات التشفير المقاومة للهجوم على الأخطاء [29].

تُنفذ دارات NCL باستخدام بوابات العتبة التي تعتبر العناصر الأولية لتحقيق منطق NCL، ويرمز لها بالرمز THmn وتسمى بوابة العتبة m-of-n، أي لها n مدخل وقيمة عتبة m حيث $n \geq m$ ، وتتميز بقدرة الاحتفاظ بالحالة أي أنه بمجرد ضبط خرج البوابة THmn يتم الاحتفاظ بحالة الخرج هذه ولن يُعاد ضبطه إلا عند إعادة ضبط عدد m على الأقل من المداخل. وبالتالي

كما هو الحال في التصميم المتزامن الذي يستخدم بوابات المنطق البوليني في تنفيذ الدارات، هنا يتم تحقيق دارات NCL باستخدام بوابات العتبة.

يوضح الشكل (2-أ) رمز بوابة العتبة TH23 كمثال عن بوابات العتبة ويبين الشكل (2-ب) بنية هذه البوابة على مستوى الترانزستور CMOS حيث تتألف من أربع كتل وظيفية هي: set-to-0، hold-0، hold-1، و set-to-1، بالإضافة إلى عاكس على الخرج مع تغذية عكسية.



الشكل (2): بوابة العتبة TH23: أ- رمزها - ب- بنيتها على مستوى الترانزستور.

6. الدراسة المرجعية:

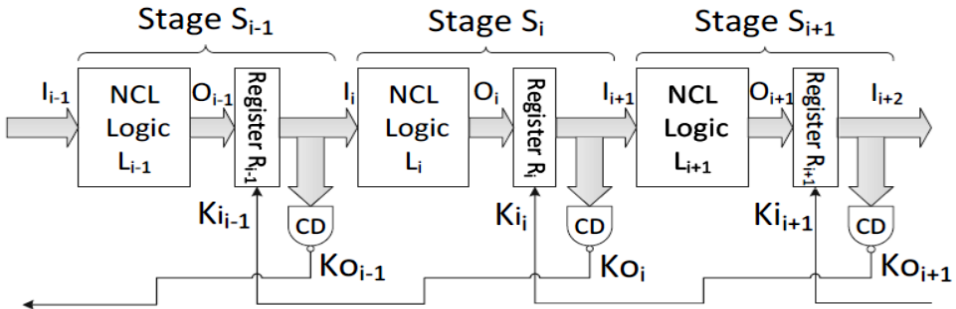
تم تنفيذ العديد من تقنيات ال pipeline غير المتزامنة باستخدام نموذج التأخير المحدود وترميز البيانات المُجمعة أحادي الأسلاك مثل MOUSETRAP و GasP و look-ahead pipeline (LP) pipeline وتعديلاتها المستندة إلى Williams PS0 pipeline بالإضافة إلى pipeline EDP (Embedded Delay Pipeline) [15-20]. يستخدم MOUSETRAP مواسك شفافة قياسية ضمن مسار التحكم وبنى منطقية ثابتة لمسار البيانات الخاص به. أيضاً تم استخدام المواسك في GasP ولكنها تتطلب مسارات تحكم متوازنة وتحتاج إلى تلبية قيود التوقيت ثنائية

الجوانب. من ناحية أخرى، تتضمن LP_{sr} pipeline مثل LP_{sr} 2/2 و LP_{sr} 2/1 مسار بيانات منطقي ديناميكي ولا يلزم وجود مواسك حيث يتم تحقيق إنتاجية أعلى في LP_{sr} 2/1 على حساب زيادة تبديد الطاقة وتعقيد التصميم بسبب الحاجة إلى تلبية ثلاثة قيود توقيت. أما بالنسبة لـ pipeline ذو التأخير المُدمج EDP، فهو يتألف من عناصر تأخير مُدمجة، وذلك للتغلب على مشاكل الـ pipelines السابقة، حيث تعمل هذه العناصر كوحدة تحكم وعنصر تأخير مُطابق بنفس الوقت، وتتطلب استيفاء قيد توقيت واحد فقط.

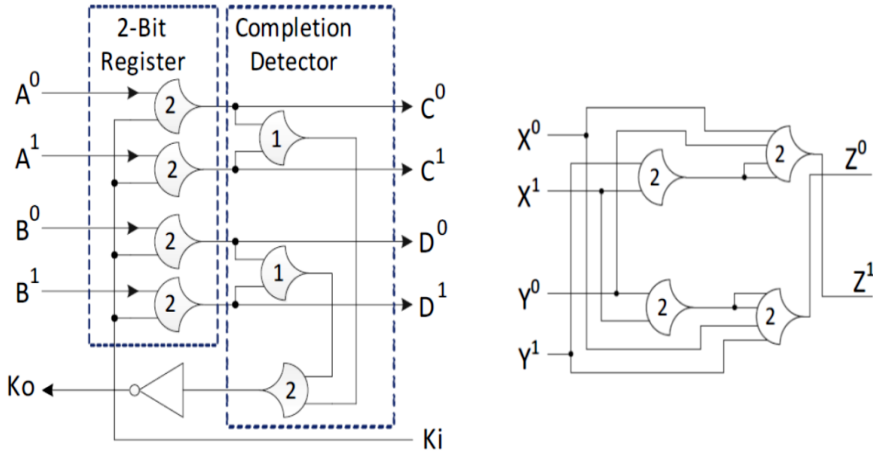
ركزت أبحاث أخرى على نماذج التصميم شبه غير حساسة للتأخير [10-14] (حيث أن تحقيق البنى غير الحساسة للتأخير تماماً صعب من الناحية العملية)، حيث تم في [13] اقتراح بنية pipeline غير حساسة للتأخير تعتمد بروتوكول ثنائي الطور مما يُتيح تقليل التأخير واستهلاك الطاقة مقارنةً ببنية QDI التي تستخدم بروتوكول رباعي الطور.

كما تم تنفيذ pipeline هجينة مثل الـ pipeline الهجين مع تأكيد الطلب المسبق (EA-Hybrid) والـ pipeline الهجين عالي السعة مع الكشف اللاحق (PD-Hybrid) [25]، التي تستخدم مسارات بيانات هجينة تجمع بين متانة الترميز ثنائي الأسلاك وبساطة مخططات الترميز أحادية السلك.

يدعم التصميم القائم على المنطق NCL أيضاً تقنية الـ pipeline [30] ويبين الشكل (3) بنية NCL pipeline التقليدية ذات ثلاث مراحل. تتكون كل مرحلة من الجزء المنطقي NCL Logic (المُنفذ باستخدام بوابات العتبة)، ومسجلات NCL لتخزين القيم المرحلية، ودارة كاشف الإكمال (Completion Detector) CD، التي تشير إلى اكتمال عملية المعالجة وتوفر بيانات جديدة.



الشكل (3): بنية NCL pipeline التقليدية.



- ب -

- أ -

الشكل (4): أ - تنفيذ دالة المنطق NCL المكافئة لبوابة XNOR بمدخلين - ب - بنية

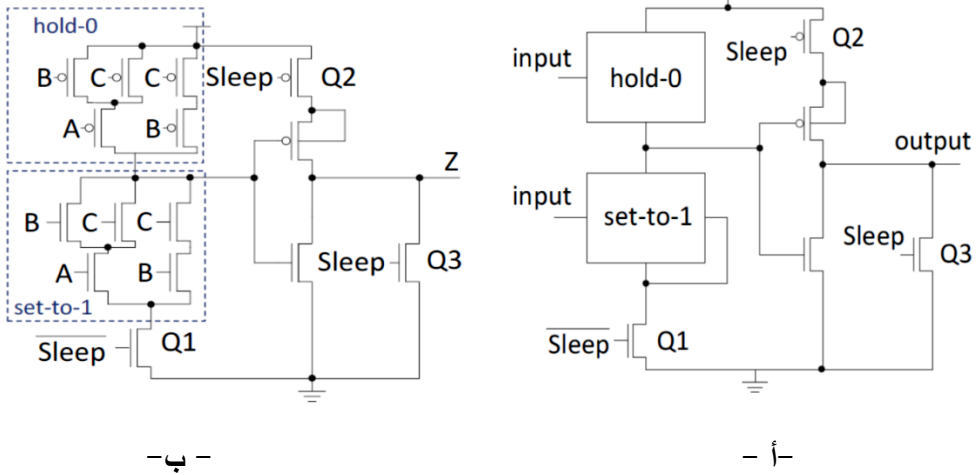
مسجل NCL وكاشف الإكمال CD الموافقين لها.

تم استخدام بوابات العتبة لتنفيذ الجزء المنطقي، والمسجلات، وكواشف الإكمال لمرحلة ال pipeline، حيث يوضح الشكل (4- أ) تنفيذ دالة منطقية تكافئ بوابة XNOR بمدخلين $n=2$ وخرج وحيد $Z = X \text{ XNOR } Y$ باستخدام بوابات العتبة كمثال عن كتلة المنطق، وتتطلب هذه الكتلة بنية مسجل NCL ثنائي البت وكاشف إكمال ثنائي البت موضحين في الشكل (4- ب). بشكل عام، يتكون مسجل NCL ثنائي البت من عدد $2n$ من بوابات العتبة TH22 ويُنفذ كاشف الإكمال ثنائي البت باستخدام من عدد n من بوابات OR ثنائية المدخل (أي TH12) وعنصر C ثنائي المدخل (أي THnn). يُستخدم كاشف الإكمال CD_i في المرحلة S_i لاستشعار ما إذا كان خرج المسجل R_i هو DATA أو NULL حيث ينتقل خرج CD_i من واحد منطقي إلى صفر منطقي عندما تأخذ جميع بنات المسجل R_i القيمة NULL والعكس صحيح.

يتألف تدفق البيانات ضمن NCL pipeline من سلسلة من موجتي NULL وDATA متناوبتين. أي أنه توجد دائماً جبهة موجة NULL/DATA بين موجتي DATA/NULL متتاليتين خلال تدفق البيانات. كما هو موضح في الشكل (3)، يتم توصيل معكوس خرج كاشف الإكمال Ko_i في المرحلة S_i بإشارة التحكم Ki_{i-1} للمسجل R_{i-1} في المرحلة السابقة S_{i-1} . عندما تمر موجة

البيانات أو موجة الـ NULL عبر الكتلة المنطقية L_i ويتم حفظها بنجاح ضمن المسجل R_i ، يصبح $K_{i-1} = 1$ و $K_0 = 0$ ، مما يُمكن المسجل R_{i-1} في المرحلة السابقة من الاحتفاظ بقيم البيانات أو قيم NULL التالية.

على الرغم من أن تصميم الـ pipeline باستخدام المنطق NCL يتميز باستهلاك طاقة منخفض كونه لا يستهلك طاقة في الحالة غير النشطة إلا أنه لا يزال يعاني من تبديد طاقة التسرب الساكنة لذلك تم اقتراح عدد من التقنيات لتصميم بوابات على مستوى الترانزستور تتميز بالتحكم الدقيق بالطاقة بهدف تقليل طاقة التسرب الساكنة في الدارات غير المتزامنة. أحد هذه التقنيات هو NCL متعدد العتبات MTNCL (Multi-threshold NCL) [33]، حيث يستخدم MTNCL ترانزستورات CMOS متعددة العتبات لتحقيق بوابات عتبة تتحكم بالطاقة بشكل دقيق، وقد تم اقتراحه أولاً في [34].

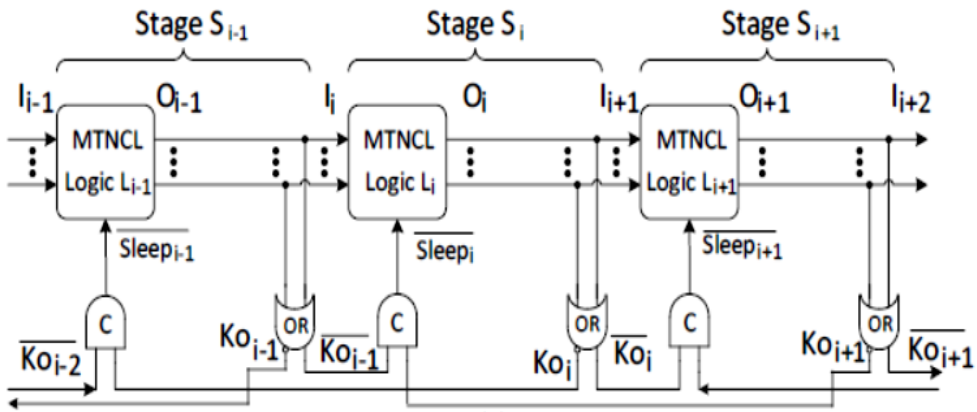


الشكل (5): أ- البنية العامة لبوابة العتبة MTNCL - ب- بنية البوابة TH23 في المنطق .MTNCL

يوضح الشكل (5-أ) البنية العامة لبوابات العتبة في المنطق MTNCL [33]، والتي تتكون من كتلتين وظيفيتين hold-0 و set-to-1، وترانزستورين سكونيين عاليي الجهد (Q1 و Q2) للتحكم الدقيق في استهلاك الطاقة بالإضافة إلى عاكس على الخرج مزود بترانزستور سحب للأسفل Q3. يظهر الشكل (5-ب) بوابة العتبة TH23 كمثال على هذه البوابات

تعمل بوابة العتبة MTNCL إما في الوضع النشط (مع إلغاء تفعيل إشارة التحكم Sleep) أو في وضع السكون (مع تفعيل إشارة التحكم Sleep). إذا كان الدخل الحالي هو DATA ويوجد صفر منطقي على مدخل إشارة التحكم $Sleep_i$ ، فيتم تشغيل الترانزستورين Q1 و Q2 في الشكل (5-أ)، وإيقاف تشغيل الترانزستور Q3 وتبدأ بوابات العتبة MTNCL بإعادة ضبط القيم على خرجها، وفي النهاية يصبح الخرج هو قيم البيانات الناتجة DATA. أما عند وجود قيمة NULL على الدخل وواحد منطقي على المدخل $Sleep_i$ ، فيتم إيقاف تشغيل الترانزستورين Q1 و Q2 مما يؤدي إلى وضع جميع بوابات العتبة بحالة خمول (بهدف الحد من استهلاك الطاقة)، وتشغيل الترانزستور Q3، وبالتالي يتم إجبار خرج جميع البوابات ليصبح صفر منطقي (أي يأخذ الخرج القيمة NULL).

استخدمت العديد من الدراسات المنطق MTNCL في تصميم الـ pipeline وتطويرها [32,31] مثل الـ MTNCL pipeline التقليدي الذي يختلف عن الـ NCL pipeline بالتصميم الداخلي لبوابات العتبة والـ Register-Less NCL pipeline (RL-NCL) الذي تم خلاله التعديل على بنية الـ MTNCL pipeline التقليدي وإلغاء جميع المسجلات، وبالتالي تقليل استهلاك المساحة والطاقة.



الشكل (6): بنية RL-NCL pipeline.

تقليل استهلاك الموارد في الشبكات ضمن الرقابة غير المتزامنة القائمة على منطق الاتفاقية الفارغة NCL

يوضح الشكل (6) بنية RL-NCL pipeline المقترح في [31]، حيث تتألف مرحلة ال pipeline المشار إليها بالرمز S_i من ثلاث مكونات: (1) كتلة منطقية Li، مُنفذة باستخدام بوابات العتبة MTNCL، (2) كاشف إكمال CD_i عبارة عن بوابة OR للكشف عما إذا كان خرج الكتلة المنطقية DATA أو NULL، (3) عنصر C، يُستخدم للتحكم في وضع التشغيل (بحالة نشطة أو سكون) للكتلة المنطقية Li فهو يولد إشارة السكون (Sleep) التي تستخدمها بوابات العتبة لإبقاء البوابة المعنية نشطة أو لوضعها بحالة سكون، وبالتالي تقليل استخدام بوابات MTNCL ضمن الكتلة المنطقية.

في [31] تمت المقارنة بين البنى الثلاثة المذكورة سابقاً وهي NCL pipeline و MTNCL pipeline و RL-NCL pipeline حيث استخدمت لتنفيذ جامع بطول 8 بت ويتطلب pipeline بخمس مراحل وأجريت هذه المحاكاة على مستوى الترانزستور باستخدام أداة المحاكاة HSPICE وتقنية 32 نانومتر.

الجدول (2): مقارنة الأداء بين NCL pipeline و MTNCL pipeline و RL-NCL

pipeline [31].

RL-NCL	MTNCL	Conv. NCL	بنية ال pipeline	
2.8	8.0	10.1	معدل دخل بيانات = 10 ميغاهرتز	استهلاك الطاقة (μW)
66.3	177.1	176.7	معدل دخل بيانات = 700 ميغاهرتز	
3132	5750	6207	عدد الترانزستورات	استهلاك
4765184	9226624	10415154	المساحة (nm^2)	الموارد

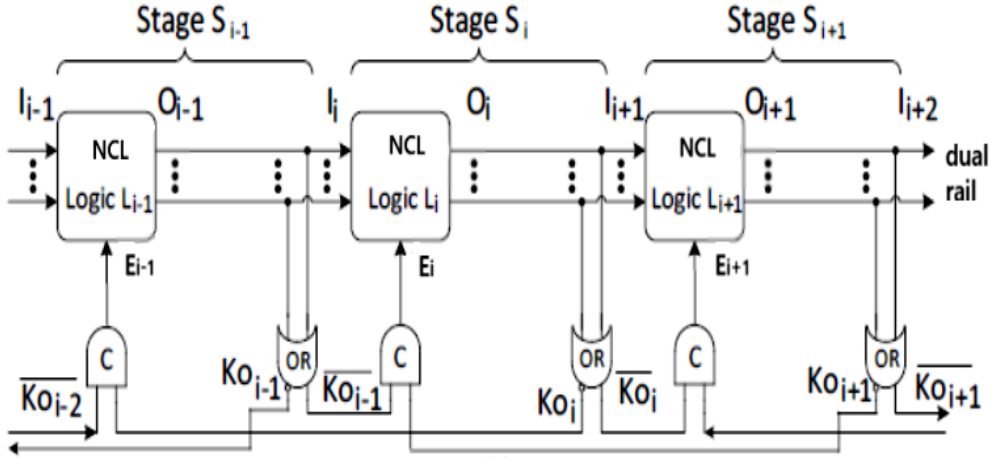
حقق RL-NCL pipeline وفقاً لنتائج المحاكاة المذكورة في الدراسة والتي تم تلخيصها في الجدول (2) استهلاكاً أقل للطاقة مقارنةً بـ NCL التقليدي سواءً عند معدلات دخل منخفضة أو عالية، وذلك بسبب أولاً التخلص من المسجلات، ثانياً استبدال كواشف الإكمال المعقدة ببوابات OR أبسط، وثالثاً التخفيف من تسرب الطاقة في الكتل المنطقية الخاملة عن طريق التحكم الدقيق

في استهلاك الطاقة. حيث يمكن لـ RL-NCL pipeline تقليل استهلاك الطاقة بنسبة 72.5% و62.5% عند معدل دخل بيانات يبلغ 10 ميغاهرتز و700 ميغاهرتز على التوالي مقارنةً بنظيره التقليدي NCL pipeline. في حين يوفر استهلاك مساحة بمقدار 45.8% وعدد ترانزستورات بمقدار 50.5%.

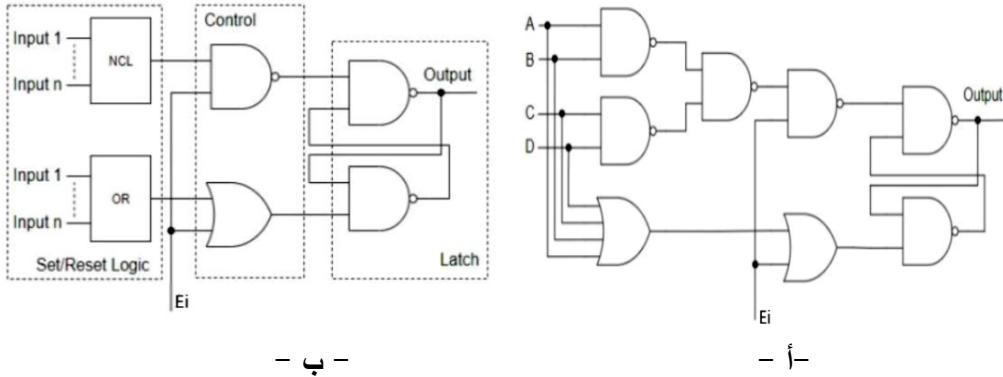
العيب الوحيد في بنية RL-NCL pipeline هو أن المشروع مخصص بالكامل تم تنفيذه على مستوى الترانزستور باستخدام بوابات المنطق MTNCL وبالتالي لا يمكن الاستفادة منه في تنفيذ التصاميم على مستوى البوابة التي تستخدم البوابات القياسية للمنطق NCL كما هي الحال في تصميمنا لموجه الشبكات ضمن الرقاقة لذلك بهدف تحسين الأداء وتقليل استهلاك الموارد في الشبكات ضمن الرقاقة القائمة على المنطق NCL ومن خلال الاستفادة من مزايا RL-NCL pipeline تم في هذا البحث اقتراح طريقة جديدة لتنفيذ بوابات NCL تغني عن حاجة RL-NCL pipeline لبوابات MTNCL وتجعله قابلاً للتنفيذ على FPGAs أو باستخدام خلايا المنطق NCL القياسية.

6. بنية pipeline المقترحة:

يوضح الشكل (7) البنية المقترحة التي تشبه إلى حد كبير بنية RL-NCL pipeline ولكن باعتماد طريقة جديدة لتنفيذ بوابات NCL ضمن الكتلة المنطقية، بحيث يتم تحقيق نفس مزايا البنية السابقة باستخدام بوابات NCL القياسية بدلاً من استخدام بوابات المنطق MTNCL التي تم تحقيقها على مستوى الترانزستور وكانت مخصصة بالكامل.



الشكل (7): بنية pipeline المقترحة.



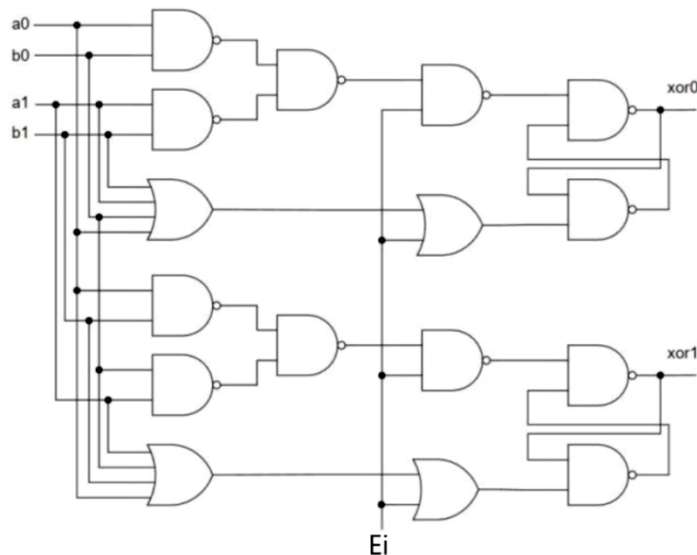
الشكل (8): طريقة التنفيذ المقترحة على مستوى البوابة: أ- البنية العامة - ب- بنية البوابة THxor0 كمثال توضيحي.

يوضح الشكل (8- أ) بنية الدارة المستخدمة لتنفيذ بوابات NCL ضمن الكتلة المنطقية للبنية المقترحة، الهدف من هذه الدارة هو محاكاة سلوك بوابات المنطق MTNCL باستخدام بوابات NCL القياسية، وهي تتألف من ثلاثة أجزاء الماسك، وكتلة التحكم، ومنطق الضبط/إعادة الضبط. الماسك هو ماسك RS قياسي يستخدم بوابات NAND، في حين تستخدم كتلة التحكم لتفعيل

الماسك RS فعندما يكون على مدخل التفعيل E_i واحد منطقي، يجب أن تُمكن الماسك من ضبط خرجه ومنع حالة إعادة الضبط، أما عندما تكون إشارة E_i صفر منطقي وجميع المداخل الأساسية بحالة NULL، يتم إعادة ضبط الماسك ومنع حالة الضبط. وبالتالي، تستخدم كتلة التحكم بوابة NAND لإشارة الضبط وبوابة OR لإشارة إعادة ضبط الماسك. تعتمد بنية الضبط وإعادة الضبط على دائرة NCL المراد تنفيذها، فمن أجل إعادة الضبط يتم استخدام بوابة OR واحدة عدد مداخلها يساوي عدد مداخل دائرة NCL المطلوبة ضمن جزء الضبط وبالتالي يكون خرج بوابة OR صفر منطقي فقط عندما يكون على جميع المداخل صفر منطقي (حالة NULL) وهذا بدوره يضمن أيضاً إلغاء حالة عدم الاستقرار من خلال إعادة ضبط الخرج فقط عندما تكون جميع المداخل صفر منطقي بينما يتمثل جزء الضبط بدالة NCL التي تعبر عن وظيفة الكتلة المنطقية.

على سبيل المثال، بفرض الكتلة المنطقية عبارة عن بوابة THxor0. يوضح الشكل (8-ب) بنية هذه الكتلة باستخدام الطريقة المقترحة، حيث تم استخدام ثلاث بوابات NAND لتنفيذ الدالة المنطقية $AB+CD$ المكافئة للبوابة THxor0 والتي تعبر عن جزء الضبط ضمن الكتلة المنطقية أما بالنسبة لإعادة الضبط فهو عبارة عن بوابة OR بأربع مداخل تبعاً لعدد المداخل الأساسية A و B و C و D للبوابة THxor0. عندما يكون على مدخل التفعيل E_i واحد منطقي، يُسمح لخرج الضبط الذي يعبر عن دالة البوابة بالوصول إلى خرج الكتلة المنطقية، وبمجرد ضبط الخرج، لن يُعاد ضبطه مرة أخرى إلا عندما يكون على المدخل E_i صفر منطقي وعلى جميع مداخل البوابة صفر منطقي أيضاً. يبين الشكل (9) البنية الموسعة لبوابة XOR بمدخلين وترميز ثنائي الأسلاك باستخدام طريقة التنفيذ المقترحة.

وهنا يجب التنويه إلى أن جميع الدارات المُنفذة تم بناءها أولاً باستخدام بوابات المنطق البوليفاني كما هو مبين في الشكلين (8,9) ثم تم تحويلها إلى شبكة من بوابات NCL المكافئة باستخدام أداة خاصة بالمنطق NCL وسوف نتطرق لذلك بالتفصيل ضمن قسم المحاكاة والنتائج خلال هذا البحث.



الشكل (9): بنية بوابة XOR بمدخلين وترميز ثنائي الأسلاك باستخدام طريقة التنفيذ المقترحة.

يتم توليد إشارة التفعيل E_i باستخدام عنصر C بمدخلين K_{0i+1} و K_{0i-1} أي أنه لا يمكن للكتلة المنطقية L_i ضمن المرحلة S_i بدء عملية معالجة بيانات جديدة D_k أو إبطال البيانات السابقة وتوليد قيمة NULL على الخرج N_k عند I_{i+1} حتى تصل قيمة NULL السابقة N_{k-1} أو قيمة البيانات السابقة D_{k-1} بأمان إلى المدخل I_{i+2} في المرحلة S_{i+2} . يمنع هذا التقييد قيم N_{k-1}/D_{k-1} الحالية من تجاوز قيم $NULL/DATA$ السابقة N_{k-1}/D_{k-1} . تبدأ جميع بوابات العتبة ضمن الكتلة المنطقية معالجة قيم البيانات الجديدة أو إبطال القيم السابقة وتوليد قيم NULL على خرجها في الوقت نفسه، بحيث يأخذ بت الخرج على المسار الحرج للكتلة المنطقية قيمة DATA أو NULL بعد أن تاخذ جميع بتات الخرج الأخرى قيم DATA أو NULL. لذلك، يمكن لمرحلة الـ pipeline المقترحة استخدام بوابة OR بمدخلين فقط لتحل محل كاشف الإكمال للكشف عما إذا كان خرج الكتلة المنطقية DATA أو NULL حيث يتصل مدخلها بزوج الأسلاك المرتبط ببيت الخرج على المسار الحرج.

يمكننا توضيح آلية عمل الـ pipeline المقترحة من خلال الخطوات التالية:

- بدايةً: بفرض أن كتلة المنطق L_i دخلت للتو في وضعية NULL (أي $E_i = 0$)، مما أدى إلى أن يصبح الخرج NULL (أي أن رمز NULL المُشار إليه بـ N_{k-1} ، موجود الآن عند O_i).
- (1) عندما تصل بيانات جديدة D_k إلى الدخل I_i للمرحلة S_i يصبح معكوس KO_{i-1} يساوي الواحد منطقي ويقوم جزء الضبط ضمن الكتلة المنطقية L_i بمعالجة هذه البيانات ولكن خرج المرحلة O_i سيبقى في وضع NULL حتى يصبح المدخل الثاني للعنصر C_{i+1} يساوي الواحد منطقي أيضاً.
- (2) بعد أن يصبح KO_{i+1} مساوياً للواحد منطقي (أي أن كتلة المنطق التالية L_{i+1} قد دخلت في وضع NULL ووصل رمز NULL السابق N_{k-1} بنجاح إلى المدخل I_{i+2})، تتغير إشارة التفعيل E_i إلى الواحد منطقي مما يسمح لخرج جزء الضبط ضمن الكتلة المنطقية بالوصول إلى خرج المرحلة O_i .
- (3) تصل رزمة البيانات الجديدة D_k إلى الدخل I_{i+1} للمرحلة التالية S_{i+1} في حين يصبح على الدخل I_i للمرحلة S_i قيم NULL مجدداً N_k ، وبالتالي يتغير معكوس KO_{i-1} إلى الصفر منطقي.
- (4) بعد أن يصبح KO_{i+1} مساوياً للصفر منطقي (أي أن قيمة البيانات السابقة D_k أصبحت على خرج الكتلة المنطقية L_{i+1} ووصلت إلى I_{i+2})، تدخل الكتلة المنطقية L_i بوضع الـ NULL ($E_i = 0$) من جديد.
- (5) في النهاية، تُكمل كتلة المنطق L_i عملية إبطال البيانات السابقة ويصبح على خرجها قيم NULL (N_k) التي تصل إلى الدخل I_{i+1} للمرحلة التالية S_{i+1} .
- (6) ويتم تكرار الخطوات السابقة مع رزم البيانات الجديدة.

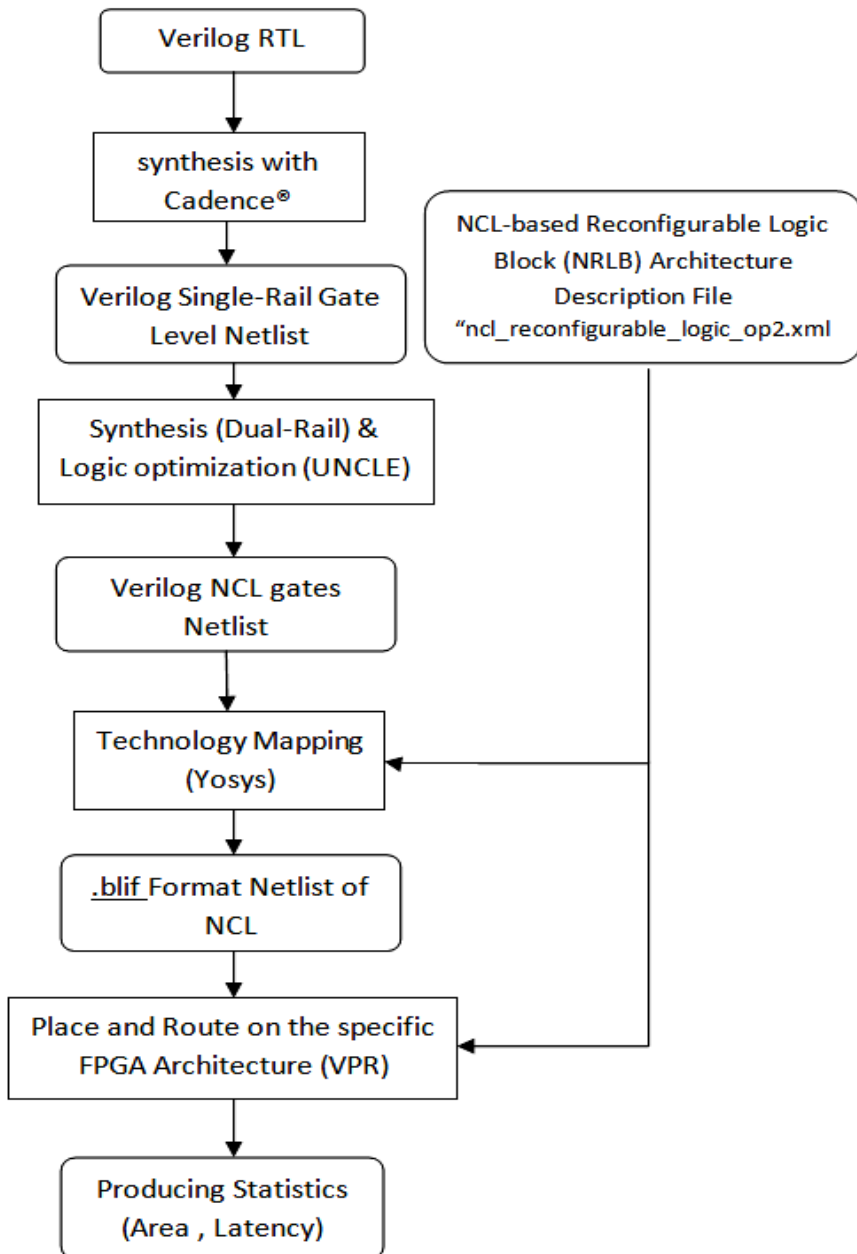
7. المحاكاة والنتائج:

1.7. تدفق التصميم والمحاكاة:

للتحقق من فعالية الطريقة المقترحة لتنفيذ بوابات NCL ضمن بنية pipeline الجديدة ودورها في تحسين أداء الشبكات ضمن الرقاقة وتقليل استهلاك الموارد، قمنا بتنفيذ موجه الشبكة باستخدام البنية المقترحة وبنية NCL pipeline التقليدية لمقارنة الأداء، وذلك من خلال الاستعانة بتدفق التصميم (Verilog-to-Routing)VTR [35] كونه يدعم التصميم غير المتزامنة ويوفر

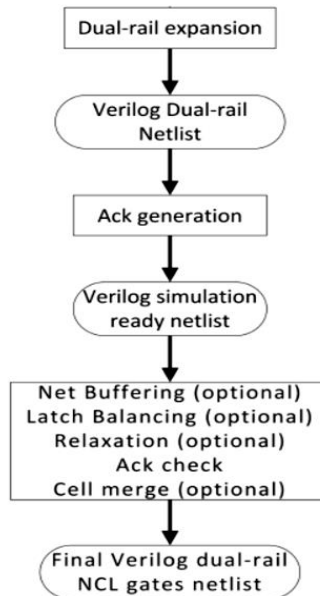
إمكانية تحقيقها على بنى قابلة لإعادة البرمجة مضمنة أو يمكن للمصمم استخدام بنية قابلة لإعادة البرمجة خارجية بإضافة ملف المواصفات الخاص بها إلى تدفق التصميم كما هو الحال في هذا البحث.

استناداً إلى تدفق تصميم CAD (Computer Aided-Design) المُخصَّص لـ VTR المبين في الشكل (10) تم بناء تصميم Verilog RTL للوحدات الفرعية لموجه NoC الموصوف في [39] باستخدام بوابات منطقية بوليانية أحادية الأسلاك من خلال أداة المحاكاة Cadence® في المرحلة الأولى. بعد ذلك، استُخدمت أداة UNCLE (Unified NCL Environment) [36]، المُبينة في الشكل (11) لتنفيذ البنية غير المتزامنة بتوليف شبكة البوابات المنطقية أحادية الأسلاك من المرحلة السابقة إلى شبكة ثنائية الأسلاك باستخدام بوابات العتبة NCL، ثم تم استخدام الأداة Yosys [37] لتنفيذ خطوة تعيين شبكة بوابات NCL على الرقابة باستخدام البنية القابلة لإعادة البرمجة NRLB (NCL based Reconfigurable Logic Block) [38] من خلال تضمين ملف المواصفات الخاص بها، وفي الخرج نحصل على ملف BLIF لبنية الدارة على مستوى البوابة. في المرحلة ما قبل الأخيرة، تُنفذ الأداة VPR (Versatile Place and Route) [35] عملية تحقيق الدارة وتوصيلها على بنية NRLB. وأخيراً، استُخرجت إحصائيات المساحة والتوقيت بواسطة VPR لتحليلها لاحقاً ومقارنتها للحصول على النتائج [35].



الشكل (10): تدفق التصميم VTR.

استناداً إلى تدفق تصميم CAD (Computer Aided-Design) المُخصَّص لـ VTR المبين في الشكل (10) تم بناء تصميم Verilog RTL للوحدات الفرعية لموجه NoC الموصوف في [39] باستخدام بوابات منطقية بوليانية أحادية الأسلاك من خلال أداة المحاكاة Cadence® في المرحلة الأولى. بعد ذلك، استُخدمت أداة UNCLE (Unified NCL Environment) [36]، المُبينة في الشكل (11) لتنفيذ البنية غير المتزامنة بتوليف شبكة البوابات المنطقية أحادية الأسلاك من المرحلة السابقة إلى شبكة ثنائية الأسلاك باستخدام بوابات العتبة NCL، ثم تم استخدام الأداة Yosys [37] لتنفيذ خطوة تعيين شبكة بوابات NCL على الرقابة باستخدام البنية القابلة لإعادة البرمجة NRLB (NCL based Reconfigurable Logic Block) [38] من خلال تضمين ملف المواصفات الخاص بها، وفي الخرج نحصل على ملف BLIF لبنية الدارة على مستوى البوابة. في المرحلة ما قبل الأخيرة، تُنفذ الأداة VPR (Versatile Place and Route) [35] عملية تحقيق الدارة وتوصيلها على بنية NRLB. وأخيراً، استُخرجت إحصائيات المساحة والتوقيت بواسطة VPR لتحليلها لاحقاً ومقارنتها للحصول على النتائج [35].

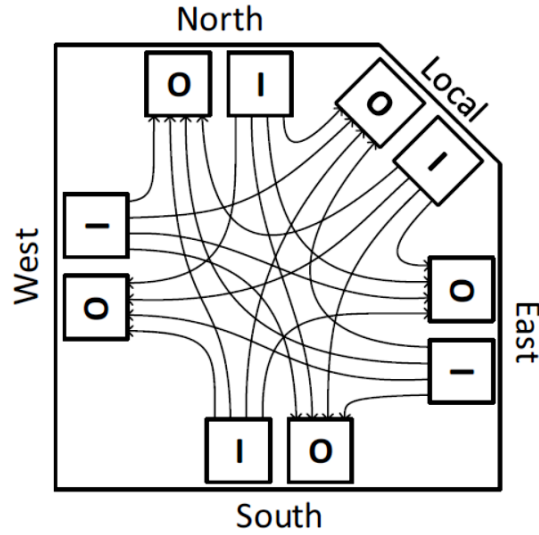


الشكل (11): تدفق تصميم أداة المحاكاة UNCLE.

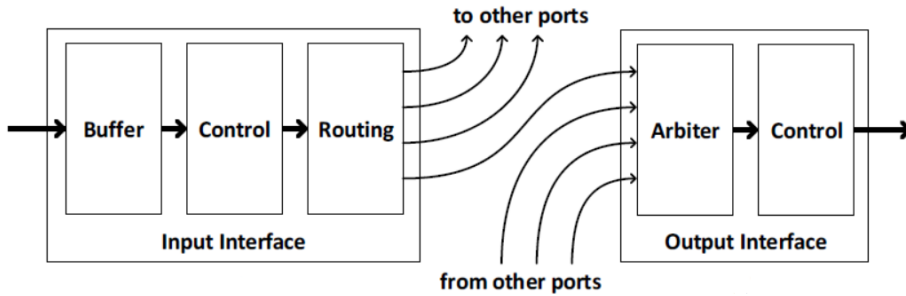
يبين الشكل (11) تدفق التصميم لأداة UNCLE [36] التي تم تنفيذها باستخدام نصوص Python® وتستخدم هذه الأداة مكتبة بوابات NCL قياسية في عملية التعيين لتحويل البوابات المنطقية البوليانية أحادية الأسلاك إلى بوابات عتبة NCL ثنائية الأسلاك، وتعطي في خرجها شبكة Verilog NCL ثنائية الأسلاك النهائية بعد إضافة شبكة المصافحة مع بعض خطوات التحسين الاختيارية.

استخدمت الكتلة المنطقية القابلة لإعادة البرمجة القائمة على المنطق NCL (NRLB) [38] كمكون للنظام القابل لإعادة البرمجة على الرقاقة، حيث تم تطويرها وتقييمها باستخدام Cadence® Virtuoso® Layout Suite مع تقنية أشباه الموصلات 45 FDSOI CMOS نانومتر. واستخدم محاكي AMS Spectre® Cadence® لمحاكاة بوابات NCL القابلة لإعادة البرمجة واستخراج نتائج تأخير البوابة التي تم توظيفها لاحقاً ضمن ملف وصف البنية المطلوب لتدفق VTR المستخدم لتقييم أداء موجه NoC.

يبين الشكلان (12) و (13) البنية المعتمدة لموجه NOC [39] وبنية أحد منافذه على التوالي وكما هو موضح في الشكل (14) تم تجزئة بنية الموجه إلى ستة مراحل pipeline تمر ضمنها رزمة البيانات لتصل إلى خرج الموجه بأمان ويتضمن الجدول (3) البارامترات الأساسية بهذا الموجه التي تم اعتمادها في عملية المحاكاة.



الشكل (12): البنية المعتمدة للموجه NOC.



الشكل (13): بنية المنفذ ضمن موجه NOC.

Stage 1	Stage 2	Stage 3	Stage 4	Stage 5	Stage 6
Buffer	Input Control	Routing	Switch Traversal	Arbiter	Output Control

الشكل (14): مراحل الـ pipeline ضمن الموجه.

الجدول (3): البارامترات الأساسية لموجه NOC المعتمدة ضمن المحاكاة [39].

Router Parameters	
Flit Data Width (bits)	32
Number of Ports	5
Buffer Depth on each port (Flits)	32
Router pipeline stages	6
Routing Algorithm	Dimension-order (XY)
Arbiter	Round-robin policy

2.7. النتائج:

تم الاستناد في النتائج إلى التقارير الزمنية وتقارير VPR الناتجة عن تدفق التصميم السابق لمقارنة أداء الموجه باستخدام NCL pipeline التقليدي وال pipeline المقترح، استخدمنا ثلاث بارامترات: (1) زمن التأخير (بالنانوثانية)، وهو الوقت اللازم لرمزة البيانات الجديدة لعبور جميع مراحل ال pipeline والوصول إلى خرج الموجه. (2) عدد الترانزستورات ذات العرض الأدنى MWTU (minimum width transistor unit) وهي الوحدة المستخدمة لقياس الموارد المستهلكة في أداة التصميم بمساعدة الحاسب الأكاديمية مفتوحة المصدر VPR (العرض الأدنى للترانزستور في تقنية أشباه الموصلات 45nm CMOS يساوي 120 nm وبالتالي مساحة الترانزستور ذو العرض الأدنى $29.952 \times 10^{-3} \text{ um}^2$). (3) عدد وحدات NRLB المستخدمة.

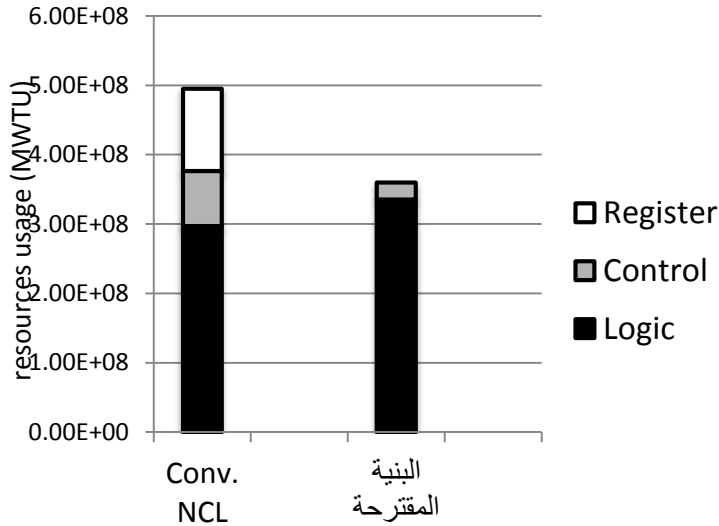
كما هو مبين في الجدول (4) فإن بنية pipeline المقترحة تحقق انخفاضاً باستهلاك موجه الشبكات ضمن الرقاقة للموارد إن كان بعدد وحدات NRLB القابلة للبرمجة المستخدمة أو بعدد الترانزستورات المقاسة بـ MWTU بنسبة 27.32% بشكل وسطي بينما يقلل زمن تأخير الموجه بنسبة 14.1% مقارنة بـ NCL pipeline التقليدي والسبب يعود إلى الطريقة المقترحة في تنفيذ بوابات NCL ضمن الكتلة المنطقية لمرحلة ال pipeline والاستفادة من بنية RL-NCL pipeline حيث وفرت هذه الطريقة تحقيق تصميم RL-NCL pipeline على مستوى البوابة

تقليل استهلاك الموارد في الشبكات ضمن الرقابة غير المتزامنة القائمة على منطق الاتفاقية الفارغة NCL

باستخدام مكتبة بوابات المنطق NCL القياسية دون الحاجة لبوابات المنطق MTNCL بالإضافة إلى التخلي عن مسجلات الـ pipeline واستبدال دائرة كاشف الإكمال المعقدة ببوابات OR بسيطة حيث يبين الشكل (15) مخطط بياني للموارد المستهلكة من قبل كل من الكتلة المنطقية ووحدة التحكم (كواشف الإكمال وعناصر C وغيرها) والمسجلات ضمن الموجه باستخدام NCL pipeline التقليدي والـ pipeline المقترح ففي التقنية التقليدية تستهلك مسجلات الـ pipeline ودائرة التحكم نحو 40% من الموارد المستخدمة ضمن الموجه في حين يبلغ استهلاك وحدة التحكم اقل من 10% من إجمالي استهلاك الموجه للموارد عند تنفيذ الـ pipeline بالطريقة المقترحة وهنا لا بد من الإشارة إلى أن هذه الطريقة تؤدي إلى زيادة بالموارد المستهلكة من قبل الكتلة المنطقية في البنية الجديدة وهي زيادة مقبولة طالما أنها تحافظ على مزايا pipeline RL-NCL وتقلل التأخير والموارد المستهلكة في الشبكات ضمن الرقابة والذي بدوره يوفر استهلاك الطاقة أيضاً.

الجدول(4): مقارنة النتائج بين الـ pipeline المقترحة و NCL pipeline التقليدي.

نسبة التحسن المئوية	البنية المقترحة	Conv. NCL	بنية الـ pipeline	
وسيطياً 27,32%	3.598568e+08	4.95125e+08	MWTU	الموارد المستهلكة
	30253	41624	عدد NRLBs	ضمن الموجه
14,1%	1193.829	1389.79	تأخير الموجه (ns)	



الشكل (15): مخطط بياني للموارد المستهلكة ضمن الموجه باستخدام البنية المقترحة والبنية التقليدية.

8. الخاتمة:

تلعب تقنية الـ pipeline في الشبكات ضمن الرقابة القائمة على المنطق NCL دوراً أساسياً في تنظيم تدفق البيانات ضمن الشبكة ولكنها تستهلك جزء كبير من الموارد لذلك تم خلال هذا البحث اقتراح بنية جديدة للـ pipeline بهدف تحسين أداء الشبكة وتقليل استهلاك الموارد انطلاقاً من البنية RL-NCL pipeline حيث تم التعديل على هذه البنية بحيث تصبح قابلة للتحقيق باستخدام بنى NCL القابلة لإعادة البرمجة من خلال اقتراح طريقة جديدة لتنفيذ بوابات NCL ضمن الكتلة المنطقية للـ pipeline بحيث تقدم سلوك مشابه لسلوك بوابات المنطق MTNCL المستخدمة ضمن البنية الأساسية للـ RL-NCL pipeline وبالتالي الاستفادة من مزايا هذه التقنية واستخدامها في تصميم بنية موجه الشبكات ضمن الرقابة غير المتزامنة القائمة على المنطق NCL حيث أثبتت نتائج المحاكاة انخفاض استهلاك الموارد من قبل موجه الشبكات ضمن الرقابة الذي يستخدم

تقليل استهلاك الموارد في الشبكات ضمن الرقابة غير المتزامنة القائمة على منطق الاتفاقية الفارغة NCL

بنية pipeline المقترحة بنسبة % 27.32 مقارنة بالبنية المُنفذة باستخدام NCL pipeline التقليدية بالإضافة إلى تقليل زمن تأخير الموجه بنسبة % 14.1 .

9. المراجع:

- [1] B. H. Calhoun, Y. Cao, X. Li, K. Mai, L. T. Pileggi, R. A. Rutenbar, and K. L. Shepard, “Digital circuit design challenges and opportunities in the era of nanoscale cmos”, Proceedings of the IEEE, vol. 96, no. 2, pp. 343–365, 2008.
- [2] J. Sparsø, “Introduction to Asynchronous Circuit Design”. DTU Compute, Technical University of Denmark, 2020.
- [3] K. M. Fant and S. A. Brandt, “NULL convention logic: A complete and consistent logic for asynchronous digital circuit synthesis”, in Proc. Int. Conf. Appl. Specific Syst. Archit. Process., pp. 261–273, 1996.
- [4] R. Parikh, R. Das, and V. Bertacco, “Power-aware NoCs through routing and topology reconfiguration”, Proceedings of the ACM/IEEE Design Automation Conference (DAC), pp.1–6, 2014.
- [5] S. M. Nowick and M. Singh, “High-performance asynchronous pipelines: An overview”, IEEE design & test of computers, vol. 28, no. 5, pp. 8–22, 2011.
- [6] S.M. Nowick and M. Singh, “Asynchronous design–Part 1: overview and recent advances”, IEEE Des. Test 32 (3), pp. 5–18, 2015.
- [7] A. Lines, “The vortex: a superscalar asynchronous processor”, Proceedings of the 13th IEEE International Symposium on Asynchronous Circuits and Systems, ASYNC, pp. 39–48, 2007.
- [8] J. Dama and A. Lines, “GHz asynchronous SRAM in 65 nm”, Proceedings of the 15th IEEE Symposium on Asynchronous Circuits and Systems, ASYNC, pp. 85–94, 2009.

- [9] J. Tse and A. Lines, “NanoMesh: an asynchronous kilo-core system-on-chip”, Proceedings of the IEEE 19th International Symposium , Asynchronous Circuits and Systems (ASYNC) May. 2013, pp. 40–49, 2013.
- [10] M. Singh and S. M. Nowick, “The design of high performance dynamic asynchronous pipelines: Lookahead style”, IEEE Trans. VLSI Syst., vol. 15, no. 11, pp. 1256–1269, 2007.
- [11] M. Singh and S. M. Nowick, “The design of high performance dynamic asynchronous pipelines: High capacity style”, IEEE Trans. VLSI Syst., vol. 15, no. 11, pp.1270–1283, 2007.
- [12] D. L. Oliveira, G. C. Duarte, N. N. M. Cardoso and G. C. Batista, “Implementation of Asynchronous Pipelines with QDI Template onto FPGAs Using Commercial Tools”, *2020 33rd Symposium on Integrated Circuits and Systems Design (SBCCI)*, Campinas, Brazil, pp. 1–5, 2020.
- [13] D. L. Oliveira, G. C. Duarte and G. C. Batista, “A New QDI Asynchronous Pipeline with Two-Phase Delay-Insensitive Global Communication”, 2021 IEEE 12th Latin America Symposium on Circuits and System (LASCAS), Arequipa, Peru, pp. 1–4, 2021.
- [14] G. C. Duarte, D. L. Oliveira and G. C. Batista, “Design of Asynchronous Pipelines with QDI Template Using Commercial FPGA”, 2022 IEEE 13th Latin America Symposium on Circuits and System (LASCAS), Puerto Varas, Chile, pp. 1–4, 2022.
- [15] M. Davies, N. Srinivasa, T. H. Lin, G. Chinya, Y. Q. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, et al., “Loihi: A neuromorphic manycore processor with on-chip learning”, IEEE Micro, vol. 38, no. 1, pp. 82–99, 2018.

- [16] A. Mardari, Z. Jel'cicov'á, and J. Sparsø, "Design and FPGAimplementation of asynchronous circuits using two-phase handshaking", in Proc. 2019 25th Int. Symp. Asynchronous Circuits and Systems, Hiroasaki, Japan, pp. 9–18, 2019.
- [17] A. Ghafoor, M. W. Mughal and A. A. Khan, "An FPGA Compliant Single-Rail Encoded Asynchronous Pipeline", IEEE Access, vol. 9, pp. 160186–160194, 2021.
- [18] S. Semba and H. Saito, "RTL Conversion Method From Pipelined Synchronous RTL Models Into Asynchronous Ones", IEEE Access, vol. 10, pp. 28949–28964, 2022.
- [19] R.K. Kavitha, A. Khajamastan and Y.R. Akhilesh et al, "High-performance asynchronous pipeline using embedded delay element", Microprocessors and Microsystems, vol 73, pp. 102955–102962, 2020.
- [20] Yu Zhou," Investigation of Asynchronous Pipeline Circuits Based on Bundled-Data Encoding: Implementation Styles, Behavioral Modeling, and Timing Analysis", TSINGHUA SCIENCE AND TECHNOLOGY, vol 27, Number 3, pp. 559–580, 2022.
- [21] P. A. Beerel, R. O. Ozdag, and M. Ferretti, "A Designer's Guide to Asynchronous VLSI", Cambridge, MA, USA :Cambridge University Press, 2010.
- [22] M. Roncken, I. Sutherland, C. Chen, Y. Hei, W. Hunt, C. Chau, S. M. Gilla, H. Park, X. Y. Song, A. P. He, et al., "How to think about self-

timed systems”, Proc. 51st Asilomar Conf. Signals, Systems, and Computers, Pacific Grove, CA, USA, pp. 1597–1604, 2017.

[23] M. Ferretti and P. A. Beerel, “High performance asynchronous design using single-track full-buffer standard cells”, IEEE J. Solid-St. Circ., vol. 41, no. 6, pp. 1444–1454, 2006.

[24] D. Edwards, W. Toms, S. Temple, L. Plana, J. Garside, and S. Furber, “The story of the amulet: A brief history of asynchronous events in Manchester”, in This Asynchronous World, Essays dedicated to Alex Yakovlev on the occasion of his 60th birthday. 2nd ed. Newcastle University, Newcastle, UK, pp. 120–130, 2017.

[25] Sravani, K., Rao, R. “Novel Asynchronous Pipeline Architectures for High-Throughput Applications”. Arab J Sci Eng 45, pp. 6625–6638, 2020.

[26] D. A. Edwards and W. B. Toms, “The status of asynchronous design in industry”, Information Society Technologies (IST) Programme, Tech. Rep. IST–1999–29119, Jun. 2004.

[27] F. A. Parsan, W. K. Al-Assadi, and S. C. Smith, “Gate mapping automation for asynchronous NULL convention logic circuits”, IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 22, no. 1, pp. 99–112, Jan. 2014.

[28] F. A. Parsan, S. C. Smith, and W. K. Al-Assadi, “Design for testability of sleep convention logic”, IEEE Trans. Very Large Scale Integ. (VLSI) Syst., vol. 24, no. 2, pp. 743–753, Feb. 2016.

- [29] Q. Ou, F. Luo, S. Li, and L. Chen, "Circuit level defines against fault attacks in pipelined NCL circuits", IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 9, pp. 1903–1913, Sep. 2015.
- [30] S. Smith and J. Di, "Designing Asynchronous Circuits using NULL Convention Logic (NCL)", 2009.
- [31] M. Chang, P. Yang, and Z. Pan, "Register-less null convention logic", IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 64, no. 3, pp. 314–318, 2017.
- [32] M. -C. Chang, M. -H. Hsieh and P. -H. Yang, "Low-Power Asynchronous NCL Pipelines With Fine-Grain Power Gating and Early Sleep," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 61, no. 12, pp. 957–961, Dec. 2014.
- [33] L. Zhou, R. Parameswaran, F. A. Parsan, S. C. Smith, and J. Di, "Multi-threshold null convention logic (mtnc1): An ultra-low power asynchronous circuit design methodology", Journal of Low Power Electronics and Applications, vol. 5, no. 2, pp. 81–100, 2015.
- [34] A. D. Bailey, J. Di, S. C. Smith, and H. A. Mantooth, "Ultra-low power delay-insensitive circuit design", in 2008 51st Midwest Symposium on Circuits and Systems, pp. 503–506, 2008.
- [35] K.E. Murray et al., "VTR 8: High-performance CAD and Customizable FPGA Architecture Modelling", ACM Transactions on Reconfigurable Technology and Systems (TRETs), vol. 13, no. 9, p 1–55, 2020.

- [36] R. B. Reese and R. A. TAYLOR. “Uncle (Unified NCL Environment)” [Online]. Available: <http://my.ece.msstate.edu/faculty/reese/uncle/UNCLE.pdf>
- [37] C. Wolf. “Yosys Open SYnthesis Suite”. Available: <http://www.clifford.at/yosys/>
- [38] J. Yu and P. Beckett, “A dual-rail LUT for reconfigurable logic using null convention logic”. In Proceedings of the 24th edition of the great lakes symposium on VLSI (GLSVLSI'14). ACM, NY, USA, pp. 261–266, 2014.
- [39] M. Moreira et al., “The YeAH! NoC Router”, Faculty of Informatics, PUCRS, Tech. Rep. 083, Dec. 2014.