Designing FPGA Based Platform for Real Time Hybrid Monitoring Network

Dr. Jamal Khalifeh[*]
Dr. Yasser Ammar[**]
Shadia Mhalla[***]

(Received 16 / 1 / 2022. Accepted 5 / 9 / 2022)

□ ABSTRACT □

In this paper, we propose a methodology to transfer various types of data simultaneously in real time via Ethernet network using UDP/IP protocols. We choose FPGA (Field Programmable Gate Array) circuits to implement the proposed methodology. We test our methodology on a surveillance platform. The platform contains monitoring camera, that can be moved in two directions: horizontal (Pan direction), and vertical (Tilt direction). The position of camera in Pan and Tilt is measured via sensors. A computer located in a geographically remote area controls the movement of the platform remotely and follows the video signal and sensors from the platform. Through Ethernet network, the signal of the cameras and sensors is transmitted from the platform to the computer, and the control signals are transmitted from the computer to the platform. The problematic in this work is resumed in two aspects; implementing UDP/IP protocols in FPGA, and guaranteeing transfer data in both directions without losses.

Keywords: FPGA, Real time, Video Signal, Ethernet, UDP/IP.

[*]**Professor -Department of Communication and Electronics- Faculty of Mechanical and Electrical Engineering - Tishreen University – Lattakia- Syria– e-mail:jam2kh58@tishreen.edu.sy**
[**]**Researcher – Scientific Studies and Research Center –Syria. yasserammar9@gmail.com**
[***]**Academic Assistant - Scientific Studies and Research Center– Syria.shadiamhalla24190@gmail.com**

تصميم منصة معتمدة على FPGA لشبكات المراقبة الهجينة بالزمن الحقيقي

د. جمال خليفة*

د. ياسر عمار **

شاديا محلا***

(تاريخ الإيداع 16 / 1 / 2022. قُبِل للنشر في 5/ 9 / 2022)

□ ملخّص □

نقدم في هذا المقال منهجية مقترحة لنقل أنماط مختلفة من البيانات بالزمن الحقيقي عبر شبكة إيثرنت(Ethernet) باستخدام بروتوكولات UDP/IP.سنعتمد الدارات القابلة للبرمجة(Field Programmable Gate Array)FPGA لإنجاز الدارات المصممة لهذه المنهجية. تم تجريب هذه المنهجية على منصة مراقبة. تحتوي المنصة على كاميرا مراقبة قابلة للتحريك بمستويين أفقي (اتجاه) وعامودي (مدى)، كما تحتوي على حساسات تدل على الوضع النسبي للكاميرا في مستويي المدى والاتجاه. يقوم حاسب موجود في منطقة بعيدة جغرافياً بالتحكم بحركة المنصة عن بعد ومتابعة إشارة الفيديو والحساسات من المنصة. يتم عبر شبكة Ethernet نقل إشارة الكاميرا والحساسات من المنصة للحاسب، ونقل إشارات التحكم من الحاسب للمنصة.يتلخص التحدي في هذا العمل بتنفيذ بروتوكولات النقل الشبكيUDP/IP باستخدام الدارات القابلة للبرمجة، إضافة لضمان نقل المعطيات بالاتجاهين دون ضياع في المعلومات.

الكلمات المفتاحية:FPGA، اشارة الفيديو، الزمن الحقيقي،Ethernet، البروتوكول UDP/IP.

*أستاذ – قسم هندسة الاتصالات والالكترونيات– كلية الهندسة الميكانيكية والكهربائية–جامعة تشرين– اللاذقية – سورية– jam2kh58@tishreen.edu.sy.

**باحث رئيسي – مركز الدراسات والبحوث العلمية – سورية–yasserammar9@gmail.com.

***قائم بالأعمال – مركز الدراسات والبحوث العلمية – سورية–shadiamhalla24190@gmail.com.

مقدمة:

قدمت الدارات القابلة للبرمجة (FPGA) حلاً وسيطاً بين مرونة المعالجات ذات الأغراض العامة وسرعة الدارات المتكاملة ذات التطبيقات الخاصة(Application Specific Integrated Circuits ASICs)، وأظهرت جدارة عالية في معالجة الكميات الكبيرة من المعطيات في الزمن الحقيقي والتي يصعب تحقيقها باستخدام المعالجات التقليدية [3][2][1].كما برزت الكفاءة العالية للدارات القابلة للبرمجة في تطبيقات معالجة الصورة الرقمية مقارنة بالمعالجات التقليدية وحتى بمعالجات الإشارة الرقمية[5][4](DSP).إضافة لذلك تقدم الدارات القابلة للبرمجة حلولاً هامة في مجال تصميم بروتوكولات الاتصال في طبقات الاتصال المختلفة.

بناءً على ما تقدم تم اعتماد الدارات القابلة للبرمجة لتصميم وتنفيذ عملية نقل المعطيات في منصة مراقبة متحكم بها عن بعد. حيث تشكل عملية نقل المعطيات المختلفة في الزمن الحقيقي تحدياً كبيراً خصوصاً إذا كانت عملية النقل تتضمن كمية كبيرة من المعلومات كإشارة الفيديو مع نقل معلومات صغيرة نسبياً بالحجم لكن تردد ورودها كبير كإشارات الحساسات.

أهمية البحث وأهدافه:

تكمن أهمية البحث في الاستفادة من الخصائص المتعددة التي تتمتع بها الدارات القابلة للبرمجة في تصميم وتنفيذ بروتوكولات الاتصال في الزمن الحقيقي على مستوى طبقة Data Link Layer (DLL). يتم استخدام بروتوكولات الاتصال في الربط الشبكي ضمن منصة مراقبة متحكم بها عن بعد. تتضمن المنصة حملاً بصرياً (كاميرا) قابل للتحريك أفقياً(Pan) وعمودياً(Tilt). تمتلك الكاميرا إمكانية تغيير التركيز (Focus) والتكبير والتصغير (Zoom). تتم كل عمليات التحكم (تحريك المحاور والتحكم بالكاميرا) من خلال حاسب بعيد جغرافياً (مركز التحكم).

لذا يقوم الحاسب بإرسال أوامر التحكم إلى المنصة عبر شبكة إيثرنت(Ethernet).كما يتم إرسال صورة الكاميرا التلفزيونية مع إشارات الحساسات الدالة على موضع الكاميرا أفقياً وعمودياً وإشارات حساسات أخرى(درجة الحرارة، عبور جسم معين وغيرها) إلى حاسب التحكم عبر شبكة إيثرنت (Ethernet).

تقوم دارة FPGAبتأمين نقل المعطيات بين الحاسب والمنصة عبر شبكة إيثرنت، وتتلخص مهامها بما يلي:

1. إرسال المعطيات المؤلفة من صورة الكاميرا الفيديوية وإشارات الحساسات (الموضع الأفقي والعمودي، درجة حرارة، عبور نطاق معين...الخ) عبر شبكة إيثرنت. يتم ذلك عن طريق تشكيل رزم المعطيات وفق بروتوكول الاتصال UDP/IP في دارة FPGA ومن ثم إرسالها إلى حاسب التحكم البعيد عبر الشبكة. يتم استقبال معطيات الحساسات في دارة FPGA بشكل تسلسلي وفق بروتوكولات النقل التسلسلي المتزامن (RS422).
2. استقبال معطيات التحكم عبر شبكة إيثرنت، ومن ثم تحويلها إلى أوامر للمحركات والكاميرا. حيث يتم تشكيل هذه المعطيات في حاسب التحكم البعيد على شكل رزم معطيات وفق بروتوكول UDP/IP وإرسالها إلى دارةFPGA عبر الشبكة. يتم إرسال أوامر التحكم من دارة FPGA بشكل تسلسلي وفق بروتوكولات النقل التسلسلي المتزامن (RS422).
3. مكاملة ومزامنة الكتل والوظائف المذكورة سابقاً (تحصيل الفيديو، إرسال واستقبال شبكي Ethernet، إرسال واستقبال تسلسلي RS422) بعدة طرق مقترحة لاختيار الحل الأمثل الذي يضمن جودة الفيديو مع تلقي أوامر التحكم ومعطيات الحساسات في الزمن الحقيقي.

طرائق البحث ومواده:

اعتماداً على المواصفات المطلوبة ولتكوين البنية الأساسية في النظام تم اختيار دارات قابلة للبرمجة ضمن بطاقة تطوير من شركة ((ALTERA(Acquired by Intel مع برنامج تطوير لدارات FPGA مجاني Quartus10[6]. بطاقة التطوير من الطراز DE2-115 والتي تحتوي بشكل أساسي على الدارة القابلة للبرمجة من النمط Cyclone®IV 4CE115، ومرمز إشارة الفيديو ADV7180 الذي يقوم بتحويل إشارة الفيديو التماثلية إلى إشارة رقمية، وموائم الشبكة إيثرنت، وموائم الوصلة التسلسلية.

لتجريب النظام، تم استخدام منصة زودت بكاميرا نهارية وأخرى حرارية، مثبتة على قاعدة ثلاثية الأرجل، زاوية الدوران الأفقي العظمى 360 درجة، زاوية الدوران العمودي العظمى 220 درجة، يتم التحكم بدوران المنصة والكاميرات عن طريق الوصلة التسلسلية RS422.

من ناحية حاسب التحكم، تم كتابة برنامج في بيئة الـ Qt يعتمد على لغة C++ لتصميم تطبيق يقوم باستلام رزم المعطيات لاستخلاص صورة الفيديو واستخلاص المعلومات المرسلة من الحساسات. يتم إظهار الفيديو وقيم الحساسات على واجهة ضمن التطبيق. كما يقوم هذا التطبيق بإرسال المعلومات التسلسلية (أوامر التحكم بالمحركات والكاميرا) وفق رزم معطيات مناسبة مشكلة وفق البروتوكول UDP/IP.

1. الدراسة المرجعية

بينت بعض الدراسات [9][8][7] الحاجة إلى أنظمة الزمن الحقيقي في معالجة الكمية الكبيرة من البيانات الملتقطة من مصادر متعددة مثل الأقمار الصناعية وأنظمة الكشف الأرضية والأجهزة الطبية ثلاثية الأبعاد وغيرها، حيث لا يمكن اعتماد المعالجات التقليدية لمعالجة كميات كبيرة من المعطيات بالزمن الحقيقي والتي تتطلب خصائص مثالية من ناحية المرونة والأداء والكلفة. اعتمدت دراسات تطبيقات معالجة الصورة الرقمية في الزمن الحقيقي [11][10] على استعراض الجدارة النسبية لدارات FPGA. حيث أظهرت هذه الدراسات خصائص هذه الدارات في معالجة الصور من حيث زيادة سرعة التنفيذ وتجنب الكثير من العمليات الزائدة عن الحاجة، والكثافة الحسابية العالية مع تكاليف تطوير منخفضة.

كما اعتمدت العديد من الدراسات المرجعية [13][12] على استخدام دارات FPGA في تطبيقات الفيديو ثلاثية الأبعاد وتنفيذها باستخدام المعالجة المتوازية، إضافة إلى التطبيقات الصناعية الشائعة الأخرى مثل PIP(Picture In Picture) و Chroma Keying. تم تقييم الفيديو المعالج باستخدام أجهزة خاصة مثل شاشات العرض الثابتة، والنظارات ثلاثية الأبعاد.

كما اهتمت بعض الدراسات [15][14] بدراسة معالجة الفيديو والصور في بيئة المعالج NIOS المصمم ضمن دارات FPGA، وظهرت قيود هذا النوع من التصميم في حالة صورة من نمط jpeg. تمثلت هذه القيود في حجم الذاكرة الأمر الذي أبرز الحاجة إلى حجوم ذاكرة إضافية وتغيير القياس الأساسي للصورة، كما ظهرت مشكلة في الكتابة إلى المخزن المؤقت للإطار في ذاكرة SRAM ضمن دارة FPGA، وذلك بسبب مشكلات المزامنة المتضمنة بداية إطار جديد من الصورة مع انشغال الذاكرة بقراءة أو كتابة إطار سابق من الصورة، إضافة إلى ظهور مشاكل متعلقة بنقل المعطيات عبر شبكة Ethernet منها فقدان بعض الرزم نتيجة عدة عوامل منها متعلقة بأنظمة التشغيل. من هنا برزت حاجتنا إلى التعامل مع صورة الفيديو بطرق جديدة لحل هذه المشاكل.

واتجهت العديد من الدراسات [16] لمواكبة النمو السريع للإنترنيت والذي تطلب مواجهة تحديين أساسيين الأول نقل المعطيات بسرعات عالية لمواكبة تطور سرعات الشبكة والذي يسمى Need for Speed، والتحدي الثاني أن يتصف

النظام بالمرونة في قابلية مواكبة التحديثات والذي يسمى Need for Flexibility. استخدم في البداية في مواجهة التحدي الأول الدارات المتكاملة ذات التطبيق الخاص(Application-Specific Integrated Circuits(ASICs التي حققت معالجة أداء عالية، لكنها فشلت في حل التحدي الثاني من ناحية المرونة. من هنا برزت أهمية FPGA في حل التحديين السابقين وتحقيق اتصالات الانترنت في الزمن الحقيقي، وذلك بسبب اتصافها بالمرونة بجزئها البرمجي والوثوقية في جزء العتاد الصلب.

واستخدمت FPGA في تصميم وظائف وبروتوكولات الاتصالات منها بروتوكول ARP(Address Resolution Protocol) والمستخدم في سبيل الحصول على العنوان الفيزيائي لعنصر الشبكة وتوثيق إنشاء الاتصال. إضافة إلى استخدام دارات FPGA في تصميم بروتوكولات الاتصال في طبقة النقل Transport Layer منها بروتوكول UDP/IP [17][18][19]، وبعض وظائف بروتوكول TCP/IP: Checksum Cyclic Redundancy Checking(CRC)،[20][21][22] Network Address Translation.

تطرقت جميع الدراسات المرجعية السابقة إلى كل من معالجة الصور وإشارات الفيديو في الزمن الحقيقي لكن بشكل مستقل عن تصميم بروتوكولات الاتصالات الخاصة بإرسال واستقبال المعطيات في الزمن الحقيقي.

وفيما يأتي سنقدم دراسة تصميمية تفصيلية للنقاط التي يعتمد عليها بحثنا.

2. الدراسة التصميمية

2.1 تصميم وتنفيذ دارة تسجيل وإرسال الفيديو

تعتمد عملية تحصيل المعطيات الكبيرة كإشارة الفيديو على دارات تقوم بتحويل إشارة الفيديو التماثلية إلى إشارة رقمية وفق ترميز خاص. نسمي هذه الدارات مرمزات الفيديو (Video Encoder).

لذا يتعين على الدارة المصممة:

1. تحصيل إشارة الفيديو من دارة مرمز الفيديو.
2. تقسيم المعطيات الخاصة بإشارة الفيديو إلى كتل معطيات قابلة للإرسال كرزم وفق البروتوكول المطلوب.
3. إرسال معطيات رزم الفيديو بالتزامن مع المعطيات الأخرى الخاصة وإشارات التحكم.

وفيما يلي شرح لهذه المراحل:

2.1.1 تحصيل إشارة الفيديو من دارة مرمز الفيديو

تتألف إشارة الفيديو من مجموعة من الصور المتتالية التي يجري عرضها بمعدل يناسب استجابة العين البشرية، مما يولد انطباعاً لمشهد واضح، غير مقطَّع، لدى الناظر. يطلق على معدل عرض الصورة، معدل الإطار frame rate. يمثل البيكسل pixel أصغر عنصر مرئي في الصورة الرقمية (الإطار Frame). يحدد موقع كل بيكسل وفقاً لثنائية (سطر، عمود). كل بيكسل هو نقطة ذات لون معين، يمثل اللون بقيمة ممثلة بعدد معين من الخانات (Bits). تعبر دقة الصورة الأفقية عن عدد البكسلات الموجودة في السطر الواحد ضمن الصورة بينما تعبر دقة الصورة العمودية عن عدد الأسطر في الصورة.

تحوي إشارة الفيديو التماثلية معلومات الصورة المتمثلة بالإضاءة Luminance واللون Color ولها عدة أنواع منها S-Video التي تحمل معلومات الإضاءة واللون في إشارات منفصلة، بينما تحمل إشارات الفيديو المركبة CVBS composite video معلومات الإضاءة واللون في إشارة واحدة. اعتمدنا في بحثنا على نوع الإشارات المركبة من نمط

NTSC و PAL اللتين تختلفان فقط بمعدل إرسال الأطر Frames وعدد أسطر الصورة كما هو موضح في الجدول 1[24][23].

الجدول 1:أنماط ترميز الفيديو

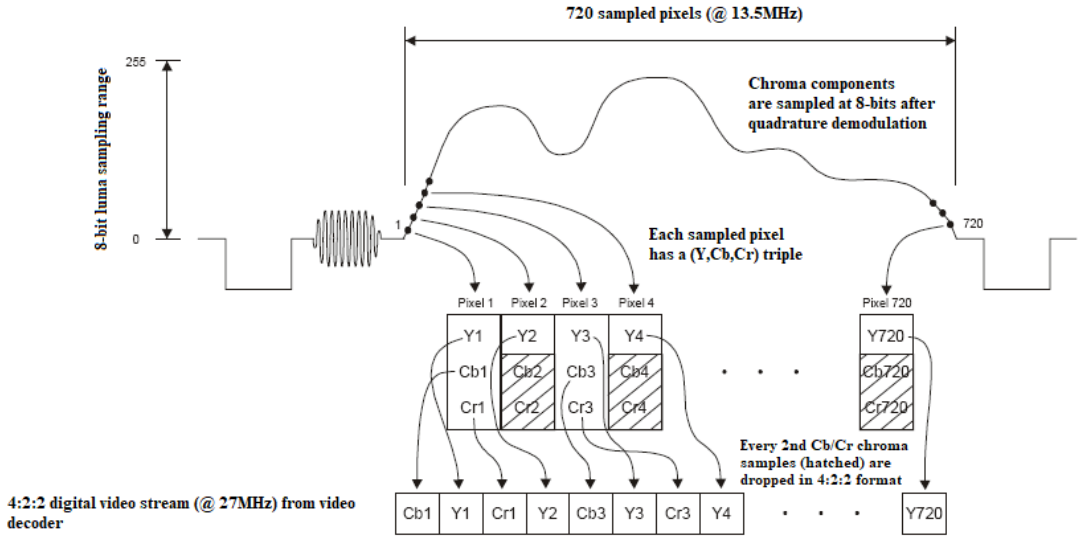
	Number ofFrames per Second	Number of Lines per Frame	Number of Active Lines
NTSC	30	525	486
PAL	50	625	576

يتم في البداية تحويل إشارة الفيديو التماثلية القادمة عبر مدخل الفيديو في بطاقة DE2-115 إلى رقمية عبر دارة متكاملة من نمط (ADV7180) والتي تقوم بتحويل نمط ترميز إشارة الفيديو من 4:4:4 إلى نمط 4:2:2 كنوع من أنواع الضغط ضمن البروتوكول ITU-RBT.656 [25].

يوجد عدة أنواع لترميز إشارة الفيديو مثل 4:4:4 و 4:2:2 و 4:1:1 و 4:2:0. يتم في النمط 4:4:4 ترميز كل بكسل من الصورة بثلاث قيم تمثل الإضاءة Y واللون Cb و Cr. يتم في نمط 4:2:2 الاحتفاظ بمعلومات الإضاءة لكل بكسل Y، بينما تختصر المعلومات اللونية الأربعة (Cb1,Cr1,Cb2,Cr2) لكل بكسلين متتاليين إلى اثنين (Cb1,Cr1)، وتسمى هذه العملية (Chroma Subsampling) هذا لأن العين البشرية حساسة للإضاءة أكثر من اللون[27][26].

يقطع كل سطر من الفيديو بتردد 13.5MHz، ينتج عن ذلك 720 بكسل في كل سطر وبما أن كل بكسلين يقابلهما أربع قيم معطيات (Y1,Cb1,Cr1,Y2)، فإن كل قيمة معطيات (Byte) يتم الحصول عليها وفق تردد 27MHz.

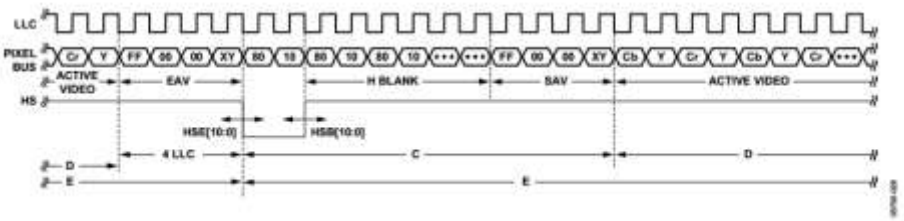
يبين **خطأ! لم يتم العثور على مصدر المرجع.**عملية التقطيع لإشارة الفيديو والترميز من نمط 4:2:2 [27][26].



الشكل 1: عملية تقطيع وترميز إشارة الفيديو إلى النمط 4:2:2 [26][27]

ويبين الشكل 1 القيم الرقمية الناتجة عن دارة ADV7180 والممثلة لنهاية سطر EAV(End Active Video) وفترة إطفاء HBLANK(Horizontal Blanking) وبداية سطر SAV(Start Active Video) مع جزء من معطيات السطر ACTIVE VIDEO [25].

Standard	Characteristic				
	HS Begin Adjust HSB[10:0] (Default)	HS End Adjust HSE[10:0] (Default)	HS to Active Video LLC Clock Cycles, C in Figure 31 (Default)	Active Video Samples/ Line, D in Figure 31	Total LLC Clock Cycles, E in Figure 31
NTSC	00000000010b	00000000000b	272	720Y + 720C = 1440	1716
PAL	00000000010b	00000000000b	284	720Y + 720C = 1440	1728



الشكل 1: القيم الرقمية الناتجة عن دارة ADV7180 [25]

حساب زمن تحصيل السطر:

تم اعتبار 1716 بايت تعبر عن سطر كامل (المعطيات مع الإطفاء)، وكل قيمة معطيات (Byte) يتم الحصول عليها وفق تردد 27MHz [27] وفق العلاقة (1) [27]:

$$t_{L,Acq} = 1716 \times \frac{1}{27 \times 10^6} = 63.5\mu s \qquad (1)$$

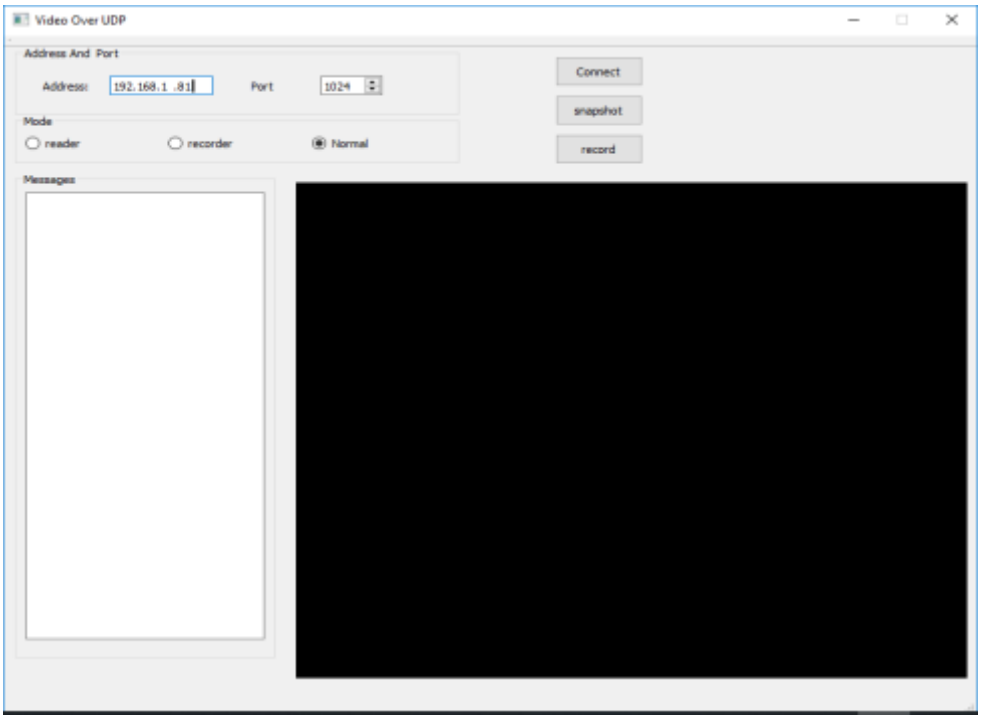
حساب زمن إرسال السطر باعتبار زمن الإرسال من أجل سرعة 100Mbps تبينه العلاقة (2)[27]:

$$t_{L,S,100Mbps} = 1400 \times \frac{1}{10^8} \times 8 = 112\mu s (2)$$

حيث تم تخفيض بايتات الإرسال من 1440 إلى 1400 بايت وذلك لأنه يوجد فاصل زمني بين الرزم المرسلة باستخدام البروتوكولات المطلوبة سواء كان UDP/IP أو TCP، ولذا استخدام 1440 بايت للإرسال سوف يؤدي إلى تداخل الرزم مع بعضها البعض وبالتالي ضياعها.

2.1.2 تقسيم المعطيات الخاصة بإشارة الفيديو إلى كتل معطيات

قمنا بتقسيم إشارة الفيديو لإرسالها عبر الشبكة وفق عدة تصاميم. تعتمد جميع هذه التصاميم على إهمال كل من الحشو الأفقي والعمودي (Blanking) ضمن أطر وأسطر الصورة وذلك من أجل توفير مساحة تخزين ضمن الذواكر المصممة في دارة FPGA. تقوم هذه الكتلة بكشف بداية كل سطر فعلي (SAV) وتوليد إشارة تفعيل تسمح بتخزين معلومات السطر الفعلي من الصورة اعتماداً على طوله. بعد تخزين أسطر الصورة الفعلية يتم إرسالها عبر الشبكة. يستقبل الحاسب المرتبط بشبكة الإيثرنت المعطيات المرسلة، ويقوم برنامج خاص (ضمن بيئة Qt) بإعادة تشكيل الصورة.يبين الشكل 2 واجهة البرنامج والجزء الخاص بإظهار الصورة.



الشكل 2: واجهة البرنامج في بيئة الـ Qt

2.1.3 بنية النظام المقترح

بارمترات التصميم:

نبين فيما يلي إشارات الدخل والخرج في دارة FPGA:

CLOCK_50: إشارة ساعة البطاقة مولدة من مهتز كريستالي بتردد 50MHz.

SW0: إشارة تبدئة (reset) للتصميم تأتي من قاطع خارجي موجود على البطاقة.

SW1, SW2: قاطعين خارجيين موجودين على البطاقة لاختيار معدل إرسال واستقبال المعلومات التسلسلية (SW1, SW2=00: معدل الإرسال (bits per second) 1200 bps، SW1, SW2=01: معدل الإرسال 9600 bps، SW1, SW2=10: معدل الإرسال 38400 bps، SW1, SW2=11: معدل الإرسال 115200bps.

KEY: مفتاح موجود على البطاقة لتفعيل دارة موائم الشبكة الموجودة على البطاقة Marvell8e1111 لإرسال القيم على خط الشبكة بسرعات محددة (10Mbps,100Mbps).

TD-CLK_27: إشارة ساعة بتردد 27MHz مولدة من دارة ADV7180.

TD-DATA: خرج دارة مرمز الفيديو ADV7180 وهي مؤلفة من ثماني خانات ومتزامنة مع إشارة TD-CLK_27.

TD_RESET_N: إشارة تفعيل مرمز الفيديو يتم إسناد قيمتها إلى 1.

UART_RXD: إشارة الاستقبال الخاصة بالتراسل التسلسليRS232.

UART_TXD: إشارة الإرسال الخاصة بالتراسل التسلسلي RS232.

ENET0_RST_N: إشارة تفعيل موائم الشبكة يتم ربطها مع مفتاح KEY.

ENET0_RX_CLK: إشارة تردد استقبال معلومات الشبكة وقيمته 25MHz.

[3:0] ENET0_RX_DATA: معلومات الأوامر القادمة عبر الشبكة من الحاسب، وذلك بعد مرورها على دارة المؤاءمة (PHY).

ENET0_RX_DV: إشارة تدل على وجود معلومات قادمة عبر الشبكة من الحاسب.

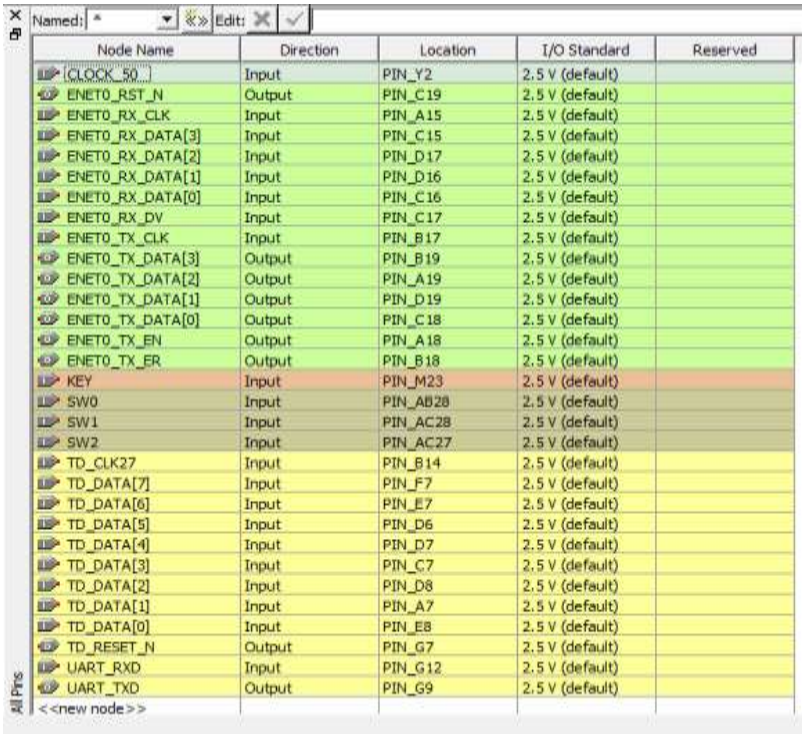
ENET0_TX_CLK: إشارة ساعة إرسال المعلومات المؤلفة من 4bits، ترددها 25MHz.

[3:0] ENET0_TX_DATA: معلومات الأوامر المرسلة عبر الشبكة من الحاسب.

ENET0_TX_EN:إشارة تفعيل الإرسال عبر الشبكة.

ENET0_TX_ER: إشارة حدوث الخطأ بالإرسال.

يبين الشكل 3 توزع مرابط الدخل والخرج Pin Assignment في دارة FPGA للبارامترات المستخدمة في بيئة الـ Quartus10 [29][28] للنظام الذي قمنا بتصميمه وتم الاستعانة بدليل الاستخدام الخاص بـ DE2-115 لمعرفة توزع الـ pins [30].



Node Name	Direction	Location	I/O Standard	Reserved
CLOCK_50	Input	PIN_Y2	2.5 V (default)	
ENET0_RST_N	Output	PIN_C19	2.5 V (default)	
ENET0_RX_CLK	Input	PIN_A15	2.5 V (default)	
ENET0_RX_DATA[3]	Input	PIN_C15	2.5 V (default)	
ENET0_RX_DATA[2]	Input	PIN_D17	2.5 V (default)	
ENET0_RX_DATA[1]	Input	PIN_D16	2.5 V (default)	
ENET0_RX_DATA[0]	Input	PIN_C16	2.5 V (default)	
ENET0_RX_DV	Input	PIN_C17	2.5 V (default)	
ENET0_TX_CLK	Input	PIN_B17	2.5 V (default)	
ENET0_TX_DATA[3]	Output	PIN_B19	2.5 V (default)	
ENET0_TX_DATA[2]	Output	PIN_A19	2.5 V (default)	
ENET0_TX_DATA[1]	Output	PIN_D19	2.5 V (default)	
ENET0_TX_DATA[0]	Output	PIN_C18	2.5 V (default)	
ENET0_TX_EN	Output	PIN_A18	2.5 V (default)	
ENET0_TX_ER	Output	PIN_B18	2.5 V (default)	
KEY	Input	PIN_M23	2.5 V (default)	
SW0	Input	PIN_AB28	2.5 V (default)	
SW1	Input	PIN_AC28	2.5 V (default)	
SW2	Input	PIN_AC27	2.5 V (default)	
TD_CLK27	Input	PIN_B14	2.5 V (default)	
TD_DATA[7]	Input	PIN_F7	2.5 V (default)	
TD_DATA[6]	Input	PIN_E7	2.5 V (default)	
TD_DATA[5]	Input	PIN_D6	2.5 V (default)	
TD_DATA[4]	Input	PIN_D7	2.5 V (default)	
TD_DATA[3]	Input	PIN_C7	2.5 V (default)	
TD_DATA[2]	Input	PIN_D8	2.5 V (default)	
TD_DATA[1]	Input	PIN_A7	2.5 V (default)	
TD_DATA[0]	Input	PIN_E8	2.5 V (default)	
TD_RESET_N	Output	PIN_G7	2.5 V (default)	
UART_RXD	Input	PIN_G12	2.5 V (default)	
UART_TXD	Output	PIN_G9	2.5 V (default)	
<<new node>>				

الشكل 3: Pin Assignment للبارامترات المستخدمة في بيئة الـ Quartus10 [30]

نذكر فيما يلي التصاميم المختلفة لتسجيل وإرسال معطيات الصورة:

2.1.3.1 التصميم الأول: مرحلة الإرسال بمعدل 10Mbps

تم تصميم الدارة التي تقوم باستقبال المعطيات الخاصة بإشارة الفيديو من المرمز ADV7180 حيث تم استخدام أربع ذواكر سعة كل منها $65536\ bytes = 2^{16}$ وهو الحجم الأعظمي الممكن توليده كذاكرة في دارة FPGA لذلك لم نتمكن من تسجيل كامل الصورة، وإنما استطعنا تسجيل جزء من الصورة (سطر كل ثلاثة أسطر) وذلك لمحدودية سعة الذواكر في البطاقة. بعد عملية التسجيل تمت قراءة محتوى الذاكرة وتضمين كل سطر في رزمة UDPوإرساله عبر الشبكة.

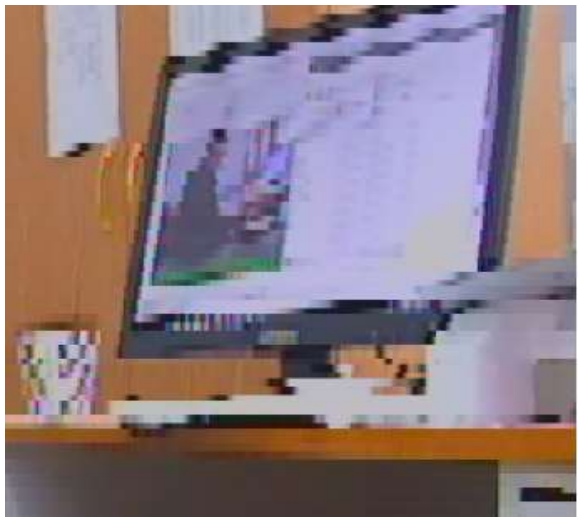
في هذه المرحلة تم تسجيل وإرسال 96 سطر زوجي و96 سطر فردي وإرسال إطار كل ثلاثة أطر. حيث أن حجم سطر الفيديو (عينات الفيديو الفعلية مع الإطفاء) 1716 بايت وكل عينة من سطر الفيديو يتم تسجيلها وفق إشارة ساعة ترددها 27MHz وبالتالي زمن ورود السطر كاملاً (المعطيات مع الإطفاء)هو $63.5\mu s$ كما ورد في العلاقة (1).
يكون حجم المعطيات في كل سطر مرسل عبر الشبكة هو 700Pixel = 1400 Bytes بتردد 27MHz وبالتالي زمن إرسال معطيات السطر:

$$t_{L,S,10Mbps} = 1400 \times \frac{1}{10^7} \times 8 = 1120\mu s \quad (3)$$

وبالتالي لدينا ضياع أكثر من 90% من المعلومات.
من أجل إظهار الفيديو المرسل عبر شبكة الإثرنيت على الحاسب، تم كتابة برنامج في بيئة Qt لتصميم تطبيق مطور يقوم باستلام رزم المعطيات وتجميعها سطر تلوالآخر لتتشكل الصورة الكاملة. حيث اعتمدنا في بيئة الـ Qtعلى انتظار ورود بداية صورة أي ورود السطر الأول من الصورة ثم قمنا بتسجيل الأسطر حسب ورودها ضمن مصفوفة الصورة وذلك بتوليد عدادين عداد يخص الأسطر الزوجية ذات البداية (80 00 00 FF) وعداد يخص الأسطر الفردية ذات البداية (C7 00 00 FF).
نقوم بفك ضغط أسطر الصورة من نمط 4:2:2 وتحويلها إلى نمط 4:4:4 بإعادة تشكيل بكسلين من كل أربع بايتات.

$$Pixel\ 1 = Cb1\ Y1\ Cr1, Pixel\ 2 = Cb1\ Y2\ Cr1 \ <== Cb1\ Y1\ Cr1\ Y2$$

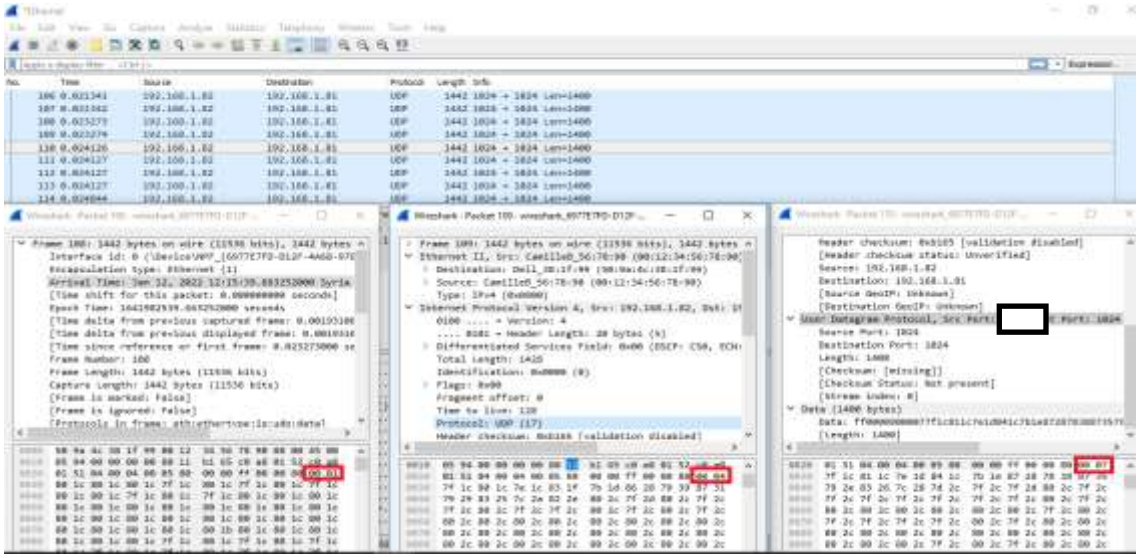
ويوضع كل بكسل في مكانه الصحيح، فكانت النتيجة الحصول على صورة غير دقيقة نتيجة اختصار الأسطر وغير مستمرة نتيجة اختصار الأطر كما هو موضح في **الشكل 4**، حيث لم نستطيع تسجيل وإرسال كامل الصورة في الزمن الحقيقي لأن زمن ورود السطر كاملاً أقل من زمن إرساله بحدود 20 مرة.



الشكل 4: لقطة من الفيديو المحصل بسرعة 10Mbps

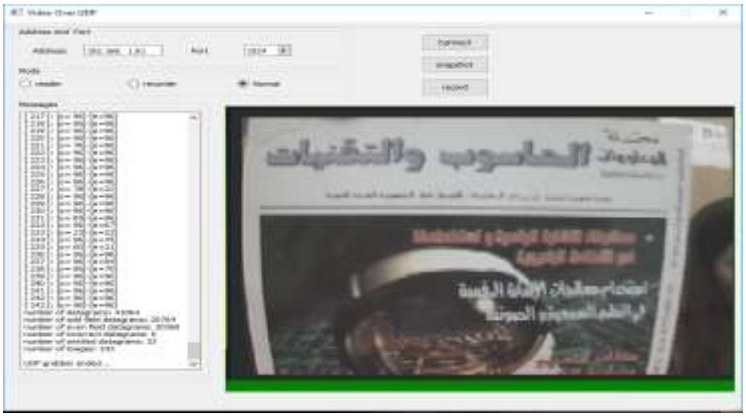
2.1.3.2 التصميم الثاني: مرحلة الإرسال بمعدل 100Mbps باستخدام ذاكرة تخزين واحدة

تم تصميم الدارة التي تقوم باستقبال المعطيات الخاصة بإشارة الفيديو من المرمز ADV7180، ومن ثم تسجيل المعطيات الموافقة لكل سطر ضمن ذاكرة FPGA وإرساله عبر الشبكة. أي يتم تقسيم الصورة إلى رزم معطيات، كل رزمة تقابل سطر فعلي من الفيديو (1400Byte) يتم تسجيله في ذاكرة وحيدة سعتها $2^{11} = 2048byte$، بتردد 27MHz. وترسل بسرعة 100Mbps وفق البروتوكول UDP/IP عبر الشبكة. حيث يبين الشكل 5 أطر UDPالمرسلة وذلك باستخدام برنامج كشف بيانات الشبكة (Wireshark) مع تبيان عدد الأسطر المرسلة (مؤطرة بشكل) حيث تم حجز بايتين بعد كل بداية FF 00 00 80, FF 00 00 C7 للإطار توضح رقم السطر المرسل.



الشكل 5:أطر UDP المرسلة على برنامج Wireshark في مرحلة الإرسال بسرعة 100Mbps بذاكرة تخزين واحدة

إن زمن إرسال سطر يعادل ثلاثة أضعاف زمن تسجيله، حيث أن زمن تحصيل سطر واحد يساوي $63.5\mu s$ بينما يبلغ زمن الإرسال من أجل سرعة 100Mbps حوالي $112\mu s$. لذا تم تسجيل وإرسال سطر وإهمال سطرين كما هو موضح من الشكل 5 (أي ما يكافئ 96 سطر فردي و96 سطر زوجي في كل إطار) ولكن دون إهمال أي إطار. كانت النتيجة صورة بنفس دقة التصميم الأول لكن بالزمن الحقيقي دون فقدان أي إطار كما هو موضح في الشكل 6 الذي يبين لقطة من الفيديو المحصل بسرعة 100Mbps وبتحصيل 96 سطر زوجي و96 سطر فردي في كل إطار.



الشكل 6:لقطة من الفيديو المحصل بسرعة 100Mbps بذاكرة تخزين واحدة

2.1.3.3 التصميم الثالث: مرحلة الإرسال بمعدل 100Mbps باستخدام ذاكرتي تخزين

تم اقتراح تحسين على التصميم الثاني، وذلك بتسجيل أسطر الفيديو الفعلية في ذاكرتين بالتناوب، وذلك بكتابة سطر في الذاكرة الأولى وقراءة سطر آخر من الذاكرة الثانية في الوقت ذاته وبالعكس. حيث تسجل معطيات الصورة على 8bit بتردد 27MHz. ومن ثم ترسل هذه المعطيات إلى دارة موائم الشبكة الموجود على البطاقة Marvell8e1111 حيث تقوم دارة المواءمة بإرسال القيم على خط الشبكة بسرعة 100Mbps. في هذا التصميم استطعنا تسجيل وإرسال (158 سطر زوجي و 158 سطر فردي فعلي من الصورة عبر البروتوكول UDP) وبالزمن الحقيقي، حيث ازداد عدد الأسطر بمقدار 60% عن التصميم الثاني وبالزمن الحقيقي. عند إظهار الصورة على الواجهة لاحظنا وجود اهتزاز واضح في الصورة نتيجة عدم ورود نفس أرقام الأسطر في كل إطار. لذلك لجأنا إلى ترقيم الأسطر القادمة من مدخل الفيديو وتسجيلها في الذاكرة وفي البرنامج المنفذ في بيئة الـ Qt رتبنا الأسطر في مصفوفة الصورة حسب رقمها وقمنا بعملية تعبئة الأسطر المفقودة في كل إطار. يوضح **خطأ! لم يتم العثور على مصدر المرجع.** أطر UDPالمرسلة.

يبين الشكل 7 لقطة من الفيديو المحصل بسرعة 100Mbps وبتحصيل 158 سطر زوجي و158 سطر فردي في كل إطار.

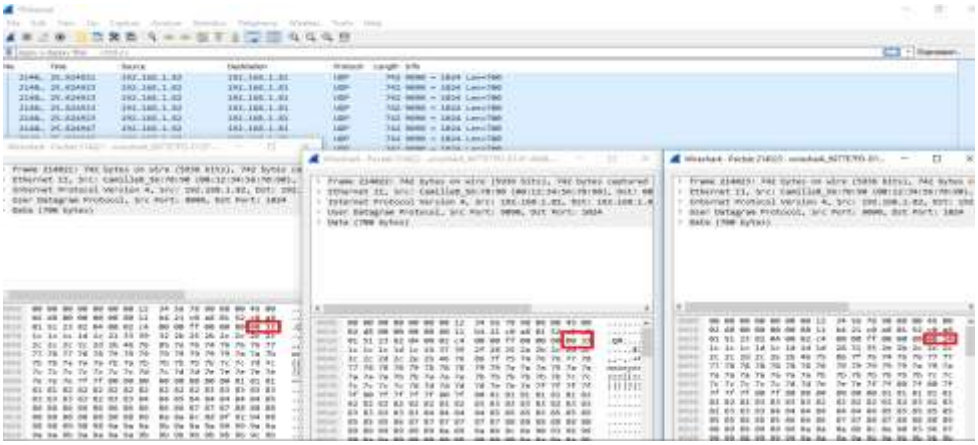
الشكل 7: لقطة من الفيديو المحصل بسرعة 100Mbps بذاكرتي تخزين

2.1.3.4 التصميم الرابع: إرسال معلومات الإضاءة للصورة فقط

من أجل تقليل حجم المعطيات المرسلة عبر الشبكة، وبالتالي تقليل زمن الإرسال، ولأن معلومات الإضاءة تعطي ما يكفي العين لتمييز الصورة ومحتوياتها، اقترحنا إرسال معطيات الإضاءة Y فقط والاستغناء عن معطيات اللون للصورة. حيث اعتمدنا التركيز على الدقة والإضاءة وعدم الاهتمام بالمعطيات اللونية لاختلافها حسب أوقات النهار والليل وعدم ضرورة الاستفادة منها.

في هذه المرحلة، قمنا بتسجيل أسطر الفيديو الفعلية في ذاكرتين بالتناوب وذلك بكتابة سطر في الذاكرة الأولى وقراءة سطر آخر من الذاكرة الثانية في الوقت ذاته وبالعكس (أي نفس المرحلة السابقة ولكن مع الأخذ بالحسبان قيم Y

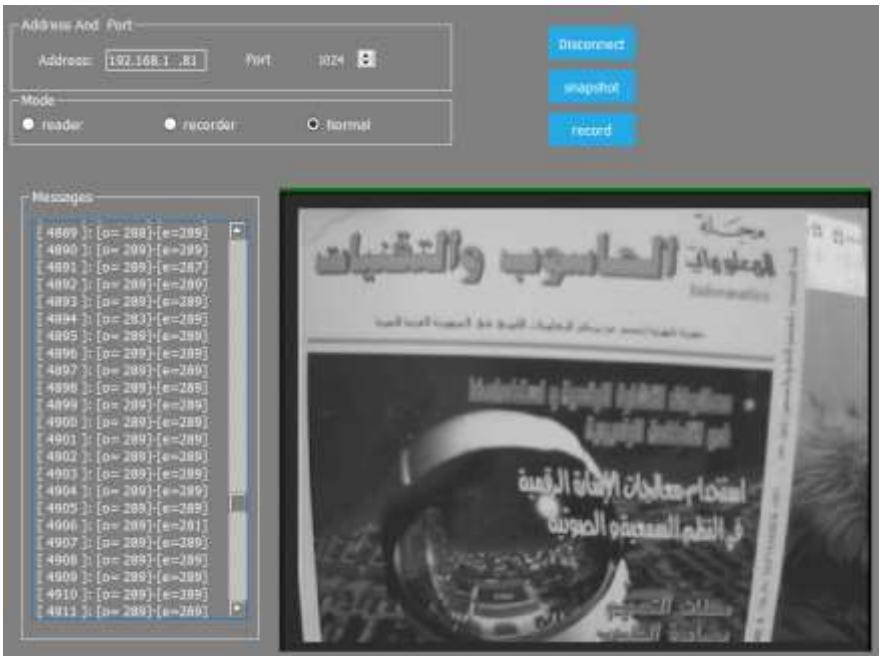
فقط). استطعنا بفضل هذا الاقتراح إرسال 288 سطر زوجي و288 سطر فردي أي استطعنا إرسال كامل الصورة بالزمن الحقيقي دون إهمال أي إطار من الفيديو. يوضح الشكل 8أطر UDPالمرسلة.



الشكل 8:أطر UDPالمرسلة على Wireshark في مرحلة الإرسال بسرعة 100Mbps تدرج رمادي

وذلك بالعودة إلى النسخة في Qtمن أجل تحصيل الصورة في هذا التصميم قمنا بتعديل الكود البرمجي في بيئة الـ سطر وإعادة 576 بايت وعدد أسطر الصورة 1400 بايت بدلاً من 700التصميم الثالث وتعديل طول السطر إلى . تم إظهار الصورة بالزمن الحقيقي Cb = 128, Cr = 128تشكيل البيكسل للحصول على اللون الرمادي بإعطاء وباللون الرمادي على الواجهة كما هو موضح في

الشكل ***9*** الذي يبين لقطة من الفيديو المحصل بسرعة 100Mbps باللون الرمادي.



الشكل 9: لقطة من الفيديو المحصل بسرعة 100Mbps تدرج رمادي

وبمقارنة هذا التصميم مع التصميم الثالث نلاحظ أنه في التصميم الثالث تعذر إرسال كامل الصورة في الزمن الحقيقي حيث تم تسجيل وإرسال 158 سطر زوجي و 158 سطر فردي فعلي من الصورة عبر البروتوكول UDP أي فقدان سطر وإرسال سطر، بينما استطعنا في التصميم الرابع إرسال 288 سطر زوجي و288 سطر فردي أي إرسال كامل الصورة وبالزمن الحقيقي دون إهمال أي إطار من الفيديو وذلك بالاكتفاء بمعلومات الإضاءة فقط لأنها تعطي ما يكفي العين لتمييز الصورة ومحتوياتها وعدم الاهتمام بالمعطيات اللونية لاختلافها حسب أوقات النهار والليل وعدم ضرورة الاستفادة منها.

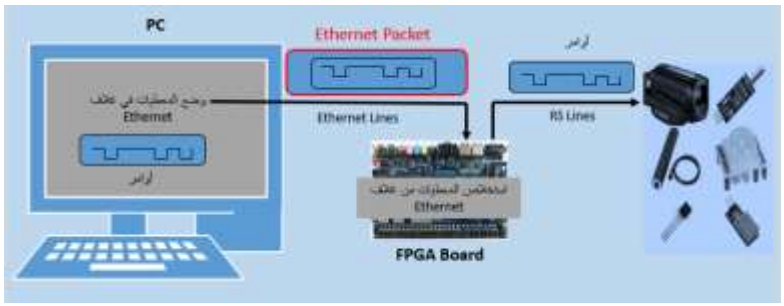
2.2 تصميم وتنفيذ دارة معالجة المعطيات التسلسلية

تم التعامل مع نوعين من المعطيات التسلسلية بين الحاسب والمنصة:

- معطيات من الحاسب إلى المنصة (أوامر تحريك المنصة والتحكم بالكاميرا)
- معطيات من المنصة إلى الحاسب (مخارج حساسات (الموضع، الظروف المحيطة)).

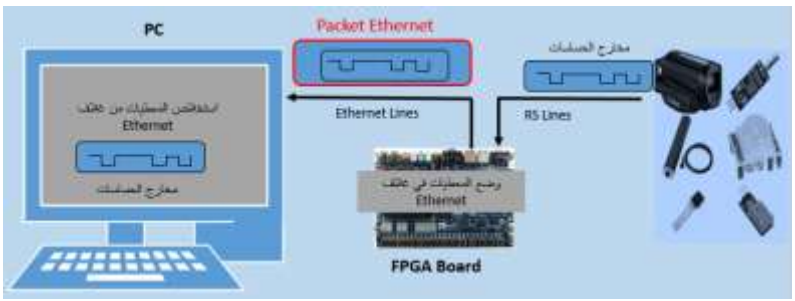
تقوم دارة FPGAبدور الوسيط لنقل هذه المعطيات عن طريق خطوط الشبكة Ethernet.

يقوم الحاسب كما هو موضح في الشكل 12 بتوليد الأوامر وتضمينها في رزمة معطيات من نوع إيثرنت Ethernet Packet، وإرسالها إلى دارة FPGA التي تقوم باستقبال هذه المعطيات، ومن ثم تحويلها إلى تدفق من المعطيات التسلسلية وفق البروتوكول RS422.



الشكل 10: نقل الأوامر من الحاسب للمنصة

من جهة أخرى تقوم دارة FPGA كما هو موضح في الشكل 13 باستقبال المعطيات التسلسلية وفق البروتوكول التسلسلي RS422، ثم تقوم بتغليف هذه المعطيات لتشكل رزمة معطيات من نوع Ethernet Packet، يتم إرسالها للحاسب.



الشكل 11: نقل إشارات الحساسات من المنصة للحاسب

لذا قمنا بتعديل التطبيق المنفذ في بيئة Qtلإرسال واستقبال رزم معطيات إيثرنت الخاصة بالأوامر والحساسات. ويبين الشكل 14 واجهة التطبيق في بيئة Qt.

الشكل 12: واجهة التطبيق في بيئة Qt

2.3 مكاملة تحصيل وإرسال الفيديو مع إرسال واستقبال المعطيات التسلسلية

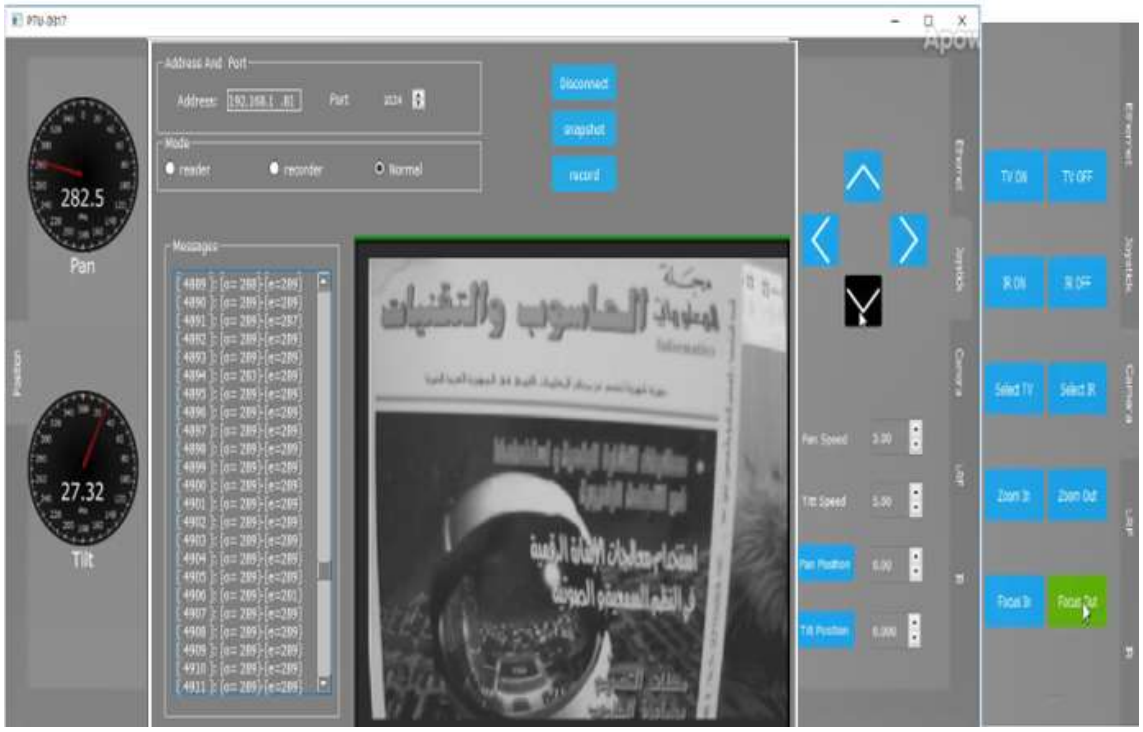
عند محاولة جمع إشارة الفيديو مع أوامر التحكم مع معطيات الحساسات ظهر تأثير سلبي على جودة الفيديو إضافة إلى حصول تأخير في أوامر التحكم، الأمر الذي أدى إلى تراكم الأوامر ومعطيات الفيديو وبالتالي خروج المنصة عن الخدمة.

لذلك ارتكز البحث على مكاملة كتلة إرسال إشارات الفيديو مع كتلة إرسال قيم الحساسات في الزمن الحقيقي، أي دمج المعلومات التسلسلية ضمن معلومات الفيديو بحيث لا تضيع معلومات الحساسات ولا تؤثر على جودة الصورة.

تم تصميم عدة نماذج من دارات الـ FPGA لتحقيق الوظائف السابقة:

- دارة تعتمد على استبدال أحد الأطر الخاص بمعطيات الفيديو(معلومات سطر)، بالمعطيات التسلسلية (مخارج الحساسات). وذلك كي لا يتم تعطيل إرسال الفيديو أثناء إرسال قيم الحساسات. تم اقتراح هذا النموذج لأن إلغاء سطر من صورة الفيديو، وليكن السطر الأول، لن يغير بشكل ملحوظ مضمون الصورة.
- دارة تعتمد على تزامن الإرسال، حيث يتم إرسال المعطيات التسلسلية في فترة الإطفاء (Blanking) والتي لا يتم فيها إرسال إشارة فيديو.

عند تطبيق الطريقتين السابقتين على التصميم الرابع الخاص بإرسال معلومات الإضاءة للصورة فقط، حصلنا على استقبال جيد لصورة الفيديو مع قيم الحساسات، وعلى إرسال أوامر التحكم من دون أي تأخير ملحوظ، كما هو موضح في الشكل 13.



الشكل 13: لقطة من الفيديو المحصل تدرج رمادي، مع قراءة لحساسات الموضع، وتحكم بالحركة والكاميرا

يبين الشكل 14 ملخص مكاملة الكتل في دارة الـ FPGA حيث أنه تم استخدام ما يقارب 2% من العناصر المنطقية الكلية وحوالي 1% من بتات الذاكرة الكلية، وهذا يظهر الحجم الكبير المتبقي في الدارة لإضافة وظائف أخرى مستقبلاً.

Flow Summary

Flow Status	Successful - Wed Dec 15 15:22:16 2021
Quartus II Version	10.0 Build 218 06/27/2010 SJ Web Edition
Revision Name	Video_and_Command
Top-level Entity Name	Video_and_Command
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Met timing requirements	N/A
Total logic elements	2,255 / 114,480 (2 %)
Total combinational functions	2,108 / 114,480 (2 %)
Dedicated logic registers	961 / 114,480 (< 1 %)
Total registers	961
Total pins	32 / 529 (6 %)
Total virtual pins	0
Total memory bits	36,980 / 3,981,312 (< 1 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)

الشكل *14*: ملخص مكاملة الكتل في دارة FPGA

الاستنتاجات والتوصيات:

تم تصميم وتنفيذ عدة نماذج من الدارات القابلة للبرمجة (FPGA) لتحقيق مكاملة ومزامنة إرسال صورة الفيديو مع المعلومات التسلسلية عبر شبكة إيثرنت وفق بروتوكول UDP/IP:

- نموذج يعتمد إرسال معلومات الصورة، وإضافة المعطيات التسلسلية المرسلة للحاسب في بداية أحد الأطر (تم كشف بداية كل إطار واستبدال معطيات الفيديو بمعطيات مخارج الحساسات). وذلك كي لا يتم تعطيل إرسال الفيديو أثناء إرسال قيم الحساسات.
- نموذج يعتمد إرسال المعطيات التسلسلية في الفترة التي لا يتم فيها إرسال إشارة فيديو، أي في زمن الإطفاء (Blanking) الأفقي أو العمودي.

أعطى النموذج الثاني (مزامنة إرسال المعلومات التسلسلية في زمن الإطفاء الخاص بالفيديو) نتائج أفضل من حيث جودة الصورة وعدم ضياع معطيات الحساسات.

يمكننا مستقبلاً استخدام خوارزميات ضغط وتشفير إشارات الفيديو والإشارات المحصلة في الزمن الحقيقي.

حيث يفيد الضغط في تقليل معطيات الصورة قدر الإمكان في محاولة تحسين عملية التحصيل في الزمن الحقيقي، كما يفيد التشفير في ضمان السرية للمعطيات المنقولة.

References:

[1] Siddiqui F, Amiri S, Minhas UI, Deng T, Woods R, Rafferty K, et al. *FPGA-based processor acceleration for image processing applications*. J Imaging. 2019;5(1):16. Available from: http://dx.doi.org/10.3390/jimaging5010016.

[2] VipinK, Fahmy SA. *FPGA dynamic and partial reconfiguration: A survey of architectures, methods, and applications*. ACM Comput Surv. 2018;51(4):1–39. Available from: http://dx.doi.org/10.1145/3193827.

[3] ASIC vs FPGA in chip design. Wevolver. 1655. Available from: https://www.wevolver.com/article/asic-vs-fpga-in-chip-design.

[4] Aissaoui N, Kaibou R, Azzaz MS. *Real-time FPGA implementation of digital video watermarking techniques using co-design approach: Comparative study*. In: 2022 7th International Conference on Image and Signal Processing and their Applications (ISPA). IEEE; 2022.

[5] Phadikar A, Mandal H, Chiu T-L. *Parallel hardware implementation of data hiding scheme for quality access control of grayscale image based on FPGA*. Multidimens Syst Signal Process. 2020;31(1):73–101. Available from: http://dx.doi.org/10.1007/s11045-019-00650-x.

[6] Parab JS, Gad RS, Naik GM. *Getting Hands on Altera® Quartus® II Software*. In: Hands-on Experience with Altera FPGA Development Boards. New Delhi: Springer India; 2017. p. 19–37.

[7] Shojima A, Yamawaki A. *Development of a Simple Verification Environment Using FPGA for image processing Hardware Created by High-Level-Synthesis Using TCP/IP*. In: The Proceedings of The 8th International Conference on Intelligent Systems and Image Processing 2021. The Institute of Industrial Applications Engineers; 2021.

[8] Nagornov NN, Lyakhov PA, Valueva MV, Bergerman MV. *RNS-Based FPGA Accelerators for High-Quality 3D Medical Image Wavelet Processing Using Scaled Filter Coefficients*. IEEE Access. 2022; 10:19215–31.

[9] Wiem B, Ahmed K, Chabha H. *Incremental Learning for Real-time Partitioning for FPGA Applications*. In: Proceedings of the 13th International Conference on Agents and Artificial Intelligence. SCITEPRESS - Science and Technology Publications; 2021.

[10] http://dx.doi.org/10.32920/ryerson.14647743.v1.

[11]http://dx.doi.org/10.1109/ispa54004.2022.9786311.

[12] Zhelnakov S. *Development and implementation of interactive 3D video environment on run-time reconfigurable FPGA platform*. Ryerson University Library and Archives; 2021 May.
[13] Nitta Y, Tamura S, Takase H. ZytleBot.*FPGA Integrated Development Platform for ROS Based Autonomous Mobile Robot*. In: 2019 29th International Conference on Field Programmable Logic and Applications (FPL). IEEE; 2019. Available from: http://dx.doi.org/10.1109/fpl.2019.00077.
[14] Meyer-Baese U. *Altera Nios Embedded Microprocessor*. In: Embedded Microprocessor System Design using FPGAs. Cham: Springer International Publishing; 2021. p. 273–337. Available from: http://dx.doi.org/10.1007/978-3-030-50533-2_9
[15] Cheng CB, Jambek AB. *Implementation of a camera system using nios II on the altera DE2-70 board*. Indonesian Journal of Electrical Engineering and Computer Science. 2019 May 1;14(2):513.
[16]Manjule H, Upadhya S, Wankhede N, Kumbhare M, Thakur K, Krishnan R. *Ethernet Implementation on FPGA*. In: 2020 International Conference for Emerging Technology (INCET). IEEE; 2020.
[17] *IMPLEMENTATION OF UDP/IP STACK IN FPGA*. International Journal of Advance Engineering and Research Development. 2015 Jul 31;2(07).
[18] Zaheer M, Khan AM. Implementation of Ethernet Data Transfer on FPGA. In: Algorithms for Intelligent Systems. Singapore: Springer Singapore; 2021. p. 131–41. Available from: http://dx.doi.org/10.1007/978-981-33-6307-6_14.
[19] http://dx.doi.org/10.1109/incet49848.2020.9154046.
[20] Chuxiong W, Haifeng S. Design and implementation of parallel CRC algorithm for *fibre channel on FPGA*. The Journal of Engineering. 2019 Oct 10;2019(21):7827–30. [21] Qaqos NN. *Optimized FPGA Implementation of the CRC Using Parallel Pipelining Architecture*. In: 2019 International Conference on Advanced Science and Engineering (ICOASE). IEEE; 2019. Available from: http://dx.doi.org/10.1109/icoase.2019.8723800.
[22] Sutter G, Ruiz M, Lopez-Buedo S, Alonso G. *FPGA-based TCP/IP Checksum Offloading Engine for 100 Gbps Networks*.In: 2018 International Conference on ReConFigurable Computing and FPGAs (ReConFig). IEEE; 2018. Available from: http://dx.doi.org/10.1109/reconfig.2018.8641729.
[23] Zhang Y-J. *Handbook of Image Engineering*. Springer Nature; 2020.
[24] *Introduction to composite NTSC and PAL*. In: Digital Video and HD. Elsevier; 2012. p. 135–9. Available from: http://dx.doi.org/10.1016/b978-0-12-391926-7.50014-x.
[25] *Inc. AD. ADV7180 10-Bit, 4× Oversampling SDTV Video Decoder Data Sheet (Rev. B)*. Available from: https://datasheet.octopart.com/ADV7180BCPZ-Analog-Devices-datasheet-129992.pdf.
[26] Jack K. *Digital Video Interfaces*. In: Video Demystified. Elsevier; 2007. p. 106–91. Available from: http://dx.doi.org/10.1016/b978-075068395-1/50006-7.
[27] Kumawat C, Pankajakshan V. *Forensics of Decompressed JPEG Color Images Based on Chroma Subsampling*. In: 2021 National Conference on Communications (NCC). IEEE; 2021. Available from:http://dx.doi.org/10.1109/ncc52529.2021.9530119.
[28] Lin M-B. *A tutorial on fpga-based system design using verilog hdl: Intel/Altera quartus version: Part I: An entry-level tutorial*. Createspace Independent Publishing
[29] http://dx.doi.org/10.1109/ncc52529.2021.9530119.
[30] *ALTERA DE2-115 USER MANUAL pdf download*. ManualsLib. 2014. Available from: https://www.manualslib.com/manual/814921/Altera-De2-115.html.